

502P0180 US00

日本国特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
with this Office

出願年月日

Date of Application:

2001年 3月29日

出願番号

Application Number:

特願2001-094803

ST.10/C ]:

[JP2001-094803]

出願人

Applicant(s):

ソニー株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2002年 2月19日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕三



【書類名】 特許願  
 【整理番号】 0100063504  
 【提出日】 平成13年 3月29日  
 【あて先】 特許庁長官殿  
 【国際特許分類】 H04L 12/28  
 G06F 19/00

## 【発明者】

【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社  
 内

【氏名】 石黒 隆二

## 【特許出願人】

【識別番号】 000002185  
 【氏名又は名称】 ソニー株式会社

## 【代理人】

【識別番号】 100082131

## 【弁理士】

【氏名又は名称】 稲本 義雄

【電話番号】 03-3369-6479

## 【手数料の表示】

【予納台帳番号】 032089

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9708842

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置および方法、記録媒体、並びにプログラム

【特許請求の範囲】

【請求項1】 コンテンツのライセンスを管理する情報処理装置において、上位の階層のノードが複数の下位の階層のノードにより構成されるツリー構造により前記ノードを規定し、前記ノードのうちの所定の階層の第1のノードに、前記コンテンツを利用するデバイスを対応させるとともに、前記ノードのうちの所定の階層の第2のノードに、前記コンテンツのライセンスを対応させ、さらに、前記第1のノードより下位の階層の第3のノードに対応するデバイスノードキーが割り当てられている前記デバイスに、前記第2のノードより下位の階層の第4のノードに対応するデバイスノードキーを供給する供給手段を備えることを特徴とする情報処理装置。

【請求項2】 前記供給手段は、所定のカテゴリの前記ライセンスに対応する1つのノードの下位の階層の複数のノードに、前記ライセンスの異なる条件をさらに対応させる

ことを特徴とする請求項1に記載の情報処理装置。

【請求項3】 コンテンツのライセンスを管理する情報処理装置の情報処理方法において、

上位の階層のノードが複数の下位の階層のノードにより構成されるツリー構造により前記ノードを規定し、前記ノードのうちの所定の階層の第1のノードに、前記コンテンツを利用するデバイスを対応させるとともに、前記ノードのうちの所定の階層の第2のノードに、前記コンテンツのライセンスを対応させ、さらに、前記第1のノードより下位の階層の第3のノードに対応するデバイスノードキーが割り当てられている前記デバイスに、前記第2のノードより下位の階層の第4のノードに対応するデバイスノードキーを供給する供給ステップを含む

ことを特徴とする情報処理方法。

【請求項4】 コンテンツのライセンスを管理する情報処理装置のプログラムにおいて、

上位の階層のノードが複数の下位の階層のノードにより構成されるツリー構造

により前記ノードを規定し、前記ノードのうちの所定の階層の第1のノードに、前記コンテンツを利用するデバイスを対応させるとともに、前記ノードのうちの所定の階層の第2のノードに、前記コンテンツのライセンスを対応させ、さらに、前記第1のノードより下位の階層の第3のノードに対応するデバイスノードキーが割り当てられている前記デバイスに、前記第2のノードより下位の階層の第4のノードに対応するデバイスノードキーを供給する供給ステップ

を含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項5】 コンテンツのライセンスを管理する情報処理装置を制御するコンピュータに、

上位の階層のノードが複数の下位の階層のノードにより構成されるツリー構造により前記ノードを規定し、前記ノードのうちの所定の階層の第1のノードに、前記コンテンツを利用するデバイスを対応させるとともに、前記ノードのうちの所定の階層の第2のノードに、前記コンテンツのライセンスを対応させ、さらに、前記第1のノードより下位の階層の第3のノードに対応するデバイスノードキーが割り当てられている前記デバイスに、前記第2のノードより下位の階層の第4のノードに対応するデバイスノードキーを供給する供給ステップ

をコンピュータに実行させるプログラム。

#### 【発明の詳細な説明】

##### 【0001】

##### 【発明の属する技術分野】

本発明は、情報処理装置および方法、記録媒体、並びにプログラムに関し、特に、異なるカテゴリにコンテンツを1つのデバイスで管理することができるようとした、情報処理装置および方法、記録媒体、並びにプログラムに関する。

##### 【0002】

##### 【従来の技術】

最近、インターネットが普及し、インターネットを介してオーディオやビデオなどの各種のコンテンツを配信することが提案され、一部、実際に実現されている。

【0003】

【発明が解決しようとする課題】

しかしながら、従来、コンテンツの著作権を管理できるようにするために、1つのデバイスに対して、1つのコンテンツだけを保持させるようにしていた。その結果、異なるカテゴリの複数のコンテンツを1つのデバイスで利用することができず、操作性が悪い課題があった。

【0004】

本発明は、このような状況に鑑みてなされたものであり、異なるカテゴリの複数のコンテンツを1つのデバイスで利用することができるようになり、操作性を改善するものである。

【0005】

【課題を解決するための手段】

本発明の情報処理装置は、上位の階層のノードが複数の下位の階層のノードにより構成されるツリー構造によりノードを規定し、ノードのうちの所定の階層の第1のノードに、コンテンツを利用するデバイスを対応させるとともに、ノードのうちの所定の階層の第2のノードに、コンテンツのライセンスを対応させ、さらに、第1のノードより下位の階層の第3のノードに対応するデバイスノードキーが割り当てられているデバイスに、第2のノードより下位の階層の第4のノードに対応するデバイスノードキーを供給する供給手段を備えることを特徴とする。

【0006】

前記供給手段は、所定のカテゴリのライセンスに対応する1つのノードの下位の階層の複数のノードに、ライセンスの異なる条件をさらに対応させるようになることができる。

【0007】

本発明の情報処理方法は、上位の階層のノードが複数の下位の階層のノードにより構成されるツリー構造によりノードを規定し、ノードのうちの所定の階層の第1のノードに、コンテンツを利用するデバイスを対応させるとともに、ノードのうちの所定の階層の第2のノードに、コンテンツのライセンスを対応させ、さ

らに、第1のノードより下位の階層の第3のノードに対応するデバイスノードキーが割り当てられているデバイスに、第2のノードより下位の階層の第4のノードに対応するデバイスノードキーを供給する供給ステップを含むことを特徴とする。

## 【0008】

本発明の記録媒体のプログラムは、上位の階層のノードが複数の下位の階層のノードにより構成されるツリー構造によりノードを規定し、ノードのうちの所定の階層の第1のノードに、コンテンツを利用するデバイスを対応させるとともに、ノードのうちの所定の階層の第2のノードに、コンテンツのライセンスを対応させ、さらに、第1のノードより下位の階層の第3のノードに対応するデバイスノードキーが割り当てられているデバイスに、第2のノードより下位の階層の第4のノードに対応するデバイスノードキーを供給する供給ステップを含むことを特徴とする。

## 【0009】

本発明のプログラムは、上位の階層のノードが複数の下位の階層のノードにより構成されるツリー構造によりノードを規定し、ノードのうちの所定の階層の第1のノードに、コンテンツを利用するデバイスを対応させるとともに、ノードのうちの所定の階層の第2のノードに、コンテンツのライセンスを対応させ、さらに、第1のノードより下位の階層の第3のノードに対応するデバイスノードキーが割り当てられているデバイスに、第2のノードより下位の階層の第4のノードに対応するデバイスノードキーを供給する供給ステップをコンピュータに実行させるプログラム。

## 【0010】

本発明の情報処理装置および方法、記録媒体、並びにプログラムにおいては、第1のノードより下位の第3のノードに対応するデバイスノードキーが割り当てられているデバイスに、第2のノードより下位の階層の第4のノードに対応するデバイスノードキーが供給される。

## 【0011】

## 【発明の実施の形態】

図1は、本発明を適用したコンテンツ提供システムの構成を示している。インターネット2には、クライアント1-1, 1-2(以下、これらのクライアントを個々に区別する必要がない場合、単にクライアント1と称する)が接続されている。この例においては、クライアントが2台のみ示されているが、インターネット2には、任意の台数のクライアントが接続される。

## 【0012】

また、インターネット2には、クライアント1に対してコンテンツを提供するコンテンツサーバ3、コンテンツサーバ3が提供するコンテンツを利用するのに必要なライセンスをクライアント1に対して付与するライセンスサーバ4、およびクライアント1がライセンスを受け取った場合に、そのクライアント1に対して課金処理を行う課金サーバ5が接続されている。

## 【0013】

これらのコンテンツサーバ3、ライセンスサーバ4、および課金サーバ5も、任意の台数、インターネット2に接続される。

## 【0014】

図2はクライアント1の構成を表している。

## 【0015】

図2において、CPU(Central Processing Unit)21は、ROM(Read Only Memory)22に記憶されているプログラム、または記憶部28からRAM(Random Access Memory)23にロードされたプログラムに従って各種の処理を実行する。タイマ20は、計時動作を行い、時刻情報をCPU21に供給する。RAM23にはまた、CPU21が各種の処理を実行する上において必要なデータなども適宜記憶される。

## 【0016】

暗号化復号部24は、コンテンツデータを暗号化するとともに、既に暗号化されているコンテンツデータを復号する処理を行う。コーデック部25は、例えば、ATRAC(Adaptive Transform Acoustic Coding)3方式などでコンテンツデータをエンコードし、入出力インターフェース32を介してドライブ30に接続されている半導体メモリ44に供給し、記録させる。あるいはまた、コーデック部2

5は、ドライブ30を介して半導体メモリ44より読み出した、エンコードされているデータをデコードする。

#### 【0017】

半導体メモリ44は、例えば、メモリストイック（商標）などにより構成される。

#### 【0018】

CPU21、ROM22、RAM23、暗号化復号部24、およびコーデック部25は、バス31を介して相互に接続されている。このバス31にはまた、入出力インターフェース32も接続されている。

#### 【0019】

入出力インターフェース32には、キーボード、マウスなどよりなる入力部26、CRT、LCDなどよりなるディスプレイ、並びにスピーカなどよりなる出力部27、ハードディスクなどより構成される記憶部28、モデム、ターミナルアダプタなどより構成される通信部29が接続されている。通信部29は、インターネット2を介しての通信処理を行う。通信部29はまた、他のクライアントとの間で、アナログ信号またはデジタル信号の通信処理を行う。

#### 【0020】

入出力インターフェース32にはまた、必要に応じてドライブ30が接続され、磁気ディスク41、光ディスク42、光磁気ディスク43、或いは半導体メモリ44などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部28にインストールされる。

#### 【0021】

なお、図示は省略するが、コンテンツサーバ3、ライセンスサーバ4、課金サーバ5も、図2に示したクライアント1と基本的に同様の構成を有するコンピュータにより構成される。そこで、以下の説明においては、図2の構成は、コンテンツサーバ3、ライセンスサーバ4、課金サーバ5などの構成としても引用される。

#### 【0022】

次に、図3のフローチャートを参照して、クライアント1がコンテンツサーバ

3からコンテンツの提供を受ける処理について説明する。

#### 【0023】

ユーザが、入力部26を操作することでコンテンツサーバ3に対するアクセスを指令すると、CPU21は、ステップS1において、通信部29を制御し、インターネット2を介してコンテンツサーバ3にアクセスさせる。ステップS2において、ユーザが、入力部26を操作して、提供を受けるコンテンツを指定すると、CPU21は、この指定情報を受け取り、通信部29から、インターネット2を介してコンテンツサーバ3に、指定されたコンテンツを通知する。図4のフローチャートを参照して後述するように、この通知を受けたコンテンツサーバ3は、暗号化されたコンテンツデータを送信してくるので、ステップS3において、CPU21は、通信部29を介して、このコンテンツデータを受信すると、ステップS4において、その暗号化されているコンテンツデータを記憶部28を構成するハードディスクに供給し、記憶させる。

#### 【0024】

次に、図4のフローチャートを参照して、クライアント1の以上の処理に対応するコンテンツサーバ3のコンテンツ提供処理について説明する。なお、以下の説明において、図2のクライアント1の構成は、コンテンツサーバ3の構成としても引用される。

#### 【0025】

ステップS21において、コンテンツサーバ3のCPU21は、インターネット2から通信部29を介してクライアント1よりアクセスを受けるまで待機し、アクセスを受けたと判定したとき、ステップS22に進み、クライアント1から送信してきたコンテンツを指定する情報を取り込む。このコンテンツを指定する情報は、クライアント1が、図3のステップS2において通知してきた情報である。

#### 【0026】

ステップS23において、コンテンツサーバ3のCPU21は、記憶部28に記憶されているコンテンツデータの中から、ステップS22の処理で取り込まれた情報で指定されたコンテンツを読み出す。CPU21は、ステップS24において

- 、記憶部28から読み出されたコンテンツデータを、暗号化復号部24に供給し
- 、コンテンツキーKcを用いて暗号化させる。

【0027】

記憶部28に記憶されているコンテンツデータは、コーデック部25により、既にATRAC3方式によりエンコードされているので、このエンコードされているコンテンツデータが暗号化されることになる。

【0028】

なお、もちろん、記憶部28に予め暗号化した状態でコンテンツデータを記憶させることができる。この場合には、ステップS24の処理は省略することが可能である。

【0029】

次に、ステップS25において、コンテンツサーバ3のCPU21は、暗号化したコンテンツデータを伝送するフォーマットを構成するヘッダに、暗号化されているコンテンツを復号するのに必要なキー情報（図5を参照して後述するEKBとEKBC（Kc））と、コンテンツを利用するのに必要なライセンスを識別するためのライセンスIDを付加する。そして、ステップS26において、コンテンツサーバ3のCPU21は、ステップS24の処理で暗号化したコンテンツと、ステップS25の処理でキーとライセンスIDを付加したヘッダとをフォーマット化したデータを、通信部29から、インターネット2を介して、アクセスしてきたクライアント1に送信する。

【0030】

図5は、このようにして、コンテンツサーバ3からクライアント1にコンテンツが供給される場合のフォーマットの構成を表している。同図に示されるように、このフォーマットは、ヘッダ（Header）とデータ（Data）とにより構成される。

【0031】

ヘッダには、コンテンツ情報（Content information）、デジタル権利管理情報（DRM（Digital Right Management）information）、ライセンスID（License ID）、イネーブリングキーブロック（有効化キーブロック）（EKB（Enabling

Key Block) ) および、 EKBから生成されたキー $K_{EKBC}$ を用いて暗号化されたコンテンツキー $Kc$ としてのデータ $K_{EKBC}(Kc)$ が配置されている。なお、 EKBについては、図15を参照して後述する。

#### 【0032】

コンテンツ情報には、データとしてフォーマット化されているコンテンツデータを識別するための識別情報としてのコンテンツID(CID)、そのコンテンツのコーデックの方式などの情報が含まれている。

#### 【0033】

デジタル権利管理情報には、コンテンツを使用する規則および状態(Usage rules/status)と、URL(Uniform Resource Locator)が配置されている。使用規則および状態には、例えば、コンテンツの再生回数、コピー回数などが記述される。

#### 【0034】

URLは、ライセンスIDで規定されるライセンスを取得するときアクセスするアドレス情報であり、図1のシステムの場合、具体的には、ライセンスを受けるために必要なライセンスサーバ4のアドレスである。ライセンスIDは、データとして記録されているコンテンツを利用するとき必要とされるライセンスを識別するものである。

#### 【0035】

データは、任意の数の暗号化ブロック(Encryption Block)により構成される。各暗号化ブロックは、イニシャルベクトル(IV(Initial Vector))、シード(Seed)、およびコンテンツデータをキー $K'c$ で暗号化したデータ $E_{K'c}(data)$ により構成されている。

#### 【0036】

キー $K'c$ は、次式により示されるように、コンテンツキー $Kc$ と、乱数で設定される値Seedをハッシュ関数に適用して演算された値により構成される。

#### 【0037】

$$K'c = \text{Hash}(Kc, \text{Seed})$$

#### 【0038】

イニシャルベクトルIVとシードSeedは、各暗号化ブロック毎に異なる値に設定される。

#### 【0039】

この暗号化は、コンテンツのデータを8バイト単位で区分して、8バイト毎に行われる。後段の8バイトの暗号化は、前段の8バイトの暗号化の結果を利用して行われるCBC(Cypher Block Chaining)モードで行われる。

#### 【0040】

CBCモードの場合、最初の8バイトのコンテンツデータを暗号化するとき、その前段の8バイトの暗号化結果が存在しないため、最初の8バイトのコンテンツデータを暗号化するときは、イニシャルベクトルIVを初期値として暗号化が行われる。

#### 【0041】

このCBCモードによる暗号化を行うことで、1つの暗号化ブロックが解読されたとしても、その影響が、他の暗号化ブロックにおよぶことが抑制される。

#### 【0042】

なお、この暗号化については、図46を参照にして、後に詳述する。

#### 【0043】

以上のようにして、クライアント1は、コンテンツサーバ3からコンテンツを無料で、自由に取得することができる。従って、コンテンツそのものは、大量に、配布することが可能となる。

#### 【0044】

しかしながら、各クライアント1は、取得したコンテンツを利用するとき、ライセンスを保持している必要がある。そこで、図6を参照して、クライアント1がコンテンツを再生する場合の処理について説明する。

#### 【0045】

ステップS41において、クライアント1のCPU21は、ユーザが入力部26を操作することで指示したコンテンツの識別情報(CID)を取得する。この識別情報は、例えば、コンテンツのタイトルや、記憶されている各コンテンツ毎に付与されている番号などにより構成される。

## 【0046】

そして、CPU21は、コンテンツが指示されると、そのコンテンツに対応するライセンスID（そのコンテンツを使用するのに必要なライセンスのID）を読み取る。このライセンスIDは、図5に示されるように、暗号化されているコンテンツデータのヘッダに記述されているものである。

## 【0047】

次に、ステップS42に進み、CPU21は、ステップS41で読み取られたライセンスIDに対応するライセンスが、クライアント1により既に取得され、記憶部28に記憶されているか否かを判定する。まだ、ライセンスが取得されていない場合には、ステップS43に進み、CPU21は、ライセンス取得処理を実行する。このライセンス取得処理の詳細は、図7のフローチャートを参照して後述する。

## 【0048】

ステップS42において、ライセンスが既に取得されていると判定された場合、または、ステップS43において、ライセンス取得処理が実行された結果、ライセンスが取得された場合、ステップS44に進み、CPU21は、取得されているライセンスは有効期限内のものであるか否かを判定する。ライセンスが有効期限内のものであるか否かは、ライセンスの内容として規定されている期限（後述する図8参照）と、タイマ20により計時されている現在日時と比較することで判断される。ライセンスの有効期限が既に満了していると判定された場合、CPU21は、ステップS45に進み、ライセンス更新処理を実行する。このライセンス更新処理の詳細は、図10のフローチャートを参照して後述する。

## 【0049】

ステップS44において、ライセンスはまだ有効期限内であると判定された場合、または、ステップS45において、ライセンスが更新された場合、ステップS46に進み、CPU21は、暗号化されているコンテンツデータを記憶部28から読み出し、RAM23に格納させる。そして、ステップS47において、CPU21は、RAM23に記憶された暗号化ブロックのデータを、図5のデータに配置されている暗号化ブロック単位で、暗号化復号部24に供給し、コンテンツキーKcを

用いて復号させる。

#### 【0050】

コンテンツキーKcを得る方法の具体例は、図15を参照して後述するが、ライセンスに含まれないデバイスノードキー(DNK)(図8)を用いて、EKB(図5)に含まれるキー $K_{EKBC}$ を得ることができ、そのキー $K_{EKBC}$ を用いて、データ $K_{EKBC}(Kc)$ (図5)から、コンテンツキーKcを得ることができる。

#### 【0051】

CPU21は、さらに、ステップS48において、暗号化復号部24により復号されたコンテンツデータをコーデック部25に供給し、デコードさせる。そして、コーデック部25によりデコードされたデータを、CPU21は、入出力インターフェース32から出力部27に供給し、D/A変換させ、スピーカから出力させる。

#### 【0052】

次に、図7のフローチャートを参照して、図6のステップS43で行われるライセンス取得処理の詳細について説明する。

#### 【0053】

最初にステップS61において、CPU21は、いま処理対象とされているライセンスIDに対応するURLを、図5に示すヘッダから取得する。上述したように、このURLは、やはりヘッダに記述されているライセンスIDに対応するライセンスを取得するときアクセスすべきアドレスである。そこで、ステップS62において、CPU21は、ステップS61で取得したURLにアクセスする。具体的には、通信部29を介してインターネット2からライセンスサーバ4にアクセスが行われる。このとき、ライセンスサーバ4は、クライアント1に対して、購入するライセンス(コンテンツを使用するのに必要なライセンス)を指定するライセンス指定情報、並びにユーザIDとパスワードの入力を要求していく(後述する図9のステップS102)。CPU21は、この要求を出力部27の表示部に表示させる。ユーザは、この表示に基づいて、入力部26を操作して、ライセンス指定情報、ユーザID、およびパスワードを入力する。なお、このユーザIDとパスワードは、クライアント1のユーザが、インターネット2を介してライセンスサーバ4にア

クセスし、事前に取得しておいたものである。

#### 【0054】

CPU21は、ステップS63, S64において、入力部26から入力されたライセンス識別情報を取り込むとともに、ユーザIDとパスワードを取り込む。CPU21は、ステップS65において、通信部29を制御し、入力されたユーザIDとパスワードを、ライセンス指定情報とともに、インターネット2を介してライセンスサーバ4に送信させる。

#### 【0055】

ライセンスサーバ4は、図9を参照して後述するように、ユーザIDとパスワード、並びにライセンス指定情報に基づいてライセンスを、そのユーザの秘密鍵(Private Key)、並びにその秘密鍵に対応する公開鍵(Public Key)の証明書とともに送信してくる(ステップS109)か、または、条件が満たされない場合には、ライセンスを送信してこない(ステップS112)。

#### 【0056】

ステップS66において、CPU21は、ライセンスサーバ4からライセンスが送信してきたか否かを判定し、ライセンスが送信されてきた場合には、ステップS67に進み、そのライセンスを、秘密鍵および公開鍵証明書とともに記憶部28に供給し、記憶させる。

#### 【0057】

ステップS66において、ライセンスが送信されて来ないと判定した場合、CPU21は、ステップS68に進み、エラー処理を実行する。具体的には、CPU21は、コンテンツを利用するためのライセンスが得られないので、コンテンツの再生処理を禁止する。

#### 【0058】

以上のようにして、各クライアント1は、コンテンツデータに付随しているライセンスIDに対応するライセンスを取得して、初めて、そのコンテンツを使用することが可能となる。

#### 【0059】

なお、図7のライセンス取得処理は、各ユーザがコンテンツを取得する前に、

予め行っておくようにすることも可能である。

#### 【0060】

クライアント1に提供されるライセンスは、例えば、図8に示されるように、使用条件、リーフIDおよびデバイスノードキー（Device Node Key (DNK)）を含んでいる。

#### 【0061】

使用条件は、そのライセンスに基づいて、コンテンツを使用することが可能な使用期限、そのライセンスに基づいて、コンテンツをダウンロードすることが可能なダウンロード期限、そのライセンスに基づいて、コンテンツをコピーすることが可能な回数（許されるコピー回数）、チェックアウト回数、最大チェックアウト回数、そのライセンスに基づいて、コンテンツをCD-Rに記録することができる権利、PD (Portable Device) にコピーすることができる回数、ライセンスを所有権（買い取り状態）に移行できる権利、使用ログをとる義務等が含まれる。

#### 【0062】

リーフIDは、Tシステム（図13を参照して後述する）により規定されるそのライセンスに割り当てられた識別情報を表し、DNKは、そのライセンスに対応するEKB（有効化キーブロック）に含まれる暗号化されているコンテンツキーKcを復号するのに必要なデバイスノードキーである（図12を参照して後述する）。

#### 【0063】

次に、図9のフローチャートを参照して、図7のクライアント1のライセンス取得処理に対応して実行されるライセンスサーバ4のライセンス提供処理について説明する。なお、この場合においても、図2のクライアント1の構成は、ライセンスサーバ4の構成として引用される。

#### 【0064】

ステップS101において、ライセンスサーバ4のCPU21は、クライアント1よりアクセスを受けるまで待機し、アクセスを受けたとき、ステップS102に進み、アクセスしてきたクライアント1に対して、ユーザIDとパスワード、並びに、ライセンス指定情報の送信を要求する。上述したようにして、クライアント1から、図7のステップS65の処理で、ユーザIDとパスワード、並びにライ

センス指定情報（ライセンスID）が送信されてきたとき、ライセンスサーバ4のCPU21は、通信部29を介してこれを受信し、取り込む処理を実行する。

#### 【0065】

そして、ライセンスサーバ4のCPU21は、ステップS103において、通信部29から課金サーバ5にアクセスし、ユーザIDとパスワードに対応するユーザの与信処理を要求する。課金サーバ5は、インターネット2を介してライセンスサーバ4から与信処理の要求を受けると、そのユーザIDとパスワードに対応するユーザの過去の支払い履歴などを調査し、そのユーザが、過去にライセンスの対価の不払いの実績があるか否かなどを調べ、そのような実績がない場合には、ライセンスの付与を許容する与信結果を送信し、不払いの実績などがある場合には、ライセンス付与の不許可の与信結果を送信する。

#### 【0066】

ステップS104において、ライセンスサーバ4のCPU21は、課金サーバ5からの与信結果が、ライセンスを付与することを許容する与信結果であるか否かを判定し、ライセンスの付与が許容されている場合には、ステップS105に進み、ステップS102の処理で取り込まれたライセンス指定情報に対応するライセンスを、記憶部28に記憶されているライセンスの中から選択する。ステップS106において、CPU21は、そのライセンスに対応して1つのDNKとリーフIDを割り当てる。さらに、ステップS107において、CPU21は、ステップS105で選択されたライセンスに対応づけられている使用条件を選択する。あるいはまた、ステップS102の処理で、ユーザから使用条件が指定された場合には、その使用条件が必要に応じて、予め用意されている使用条件に付加される。

#### 【0067】

ステップS108において、CPU21は、ステップS107で選択した使用条件を、ステップS106で割り当てたDNKとリーフIDに対応付けて付加する。これにより、図8に示されるような構成のライセンスが生成される。

#### 【0068】

次に、ステップS109に進み、ライセンスサーバ4のCPU21は、そのライセンス（図8に示される構成を有する）を、そのクライアント1のユーザに割り

当てる秘密鍵、並びに、その秘密鍵に対応する公開鍵の証明書とともに、通信部29からインターネット2を介してクライアント1に送信させる。但し、秘密鍵と証明書は、そのユーザに1つ割り当てられるのもあり、ライセンス毎に異なるものではないので、既に、送付されている場合には、省略することができる。

## 【0069】

ステップS110においてライセンスサーバ4のCPU21は、ステップS109の処理で、いま送信したライセンス（使用条件、リーフID、およびDNKを含む）を、証明書および秘密鍵とともに、ステップS102の処理で取り込まれたユーザIDとパスワードに対応して、記憶部28に記憶させる。さらに、ステップS111において、CPU21は、課金処理を実行する。具体的には、CPU21は、通信部29から課金サーバ5に、そのユーザIDとパスワードに対応するユーザに対する課金処理を要求する。課金サーバ5は、この課金の要求に基づいて、そのユーザに対する課金処理を実行する。上述したように、この課金処理に対して、そのユーザが支払いを行わなかったような場合には、以後、そのユーザは、ライセンスの付与を要求したとしても、ライセンスを受けることができないことになる。

## 【0070】

すなわち、この場合には、課金サーバ5からライセンスの付与を不許可とする与信結果が送信されてくるので、ステップS104からステップS112に進み、CPU21は、エラー処理を実行する。具体的には、ライセンスサーバ4のCPU21は、通信部29を制御してアクセスしてきたクライアント1に対して、ライセンスを付与することができない旨のメッセージを出力し、処理を終了させる。

## 【0071】

この場合、上述したように、そのクライアント1はライセンスを受けることができないので、そのコンテンツを利用すること（暗号を復号すること）ができないことになる。

## 【0072】

図10は、図6のステップS45におけるライセンス更新処理の詳細を表している。図10のステップS131乃至ステップS135の処理は、図7のステッ

プS61乃至ステップS65の処理と基本的に同様の処理である。ただし、ステップS133において、CPU21は、購入するライセンスではなく、更新するライセンスのライセンスIDを取り込む。そして、ステップS135において、CPU21は、ユーザIDとパスワードとともに、更新するライセンスのライセンスIDを、ライセンスサーバ4に送信する。

## 【0073】

ステップS135の送信処理に対応して、ライセンスサーバ4は、後述するように、使用条件を提示してくる（図11のステップS153）。そこで、クライアント1のCPU21は、ステップS136において、ライセンスサーバ4からの使用条件の提示を受信し、これを出力部27に出力し、表示させる。ユーザは、入力部26を操作して、この使用条件の中から所定の使用条件を選択したり、所定の使用条件を新たに追加したりする。ステップS137でCPU21は、以上のようにして選択された使用条件（ライセンスを更新する条件）を購入するための申し込みをライセンスサーバ4に送信する。この申し込みに対応して、後述するようにライセンスサーバ4は、最終的な使用条件を送信してくる（図11のステップS154）。そこで、ステップS138において、クライアント1のCPU21は、ライセンスサーバ4からの使用条件を取得し、ステップS139において、その使用条件を記憶部28にすでに記憶されている対応するライセンスの使用条件として更新する。

## 【0074】

図11は、以上のクライアント1のライセンス更新処理に対応して、ライセンスサーバ4が実行するライセンス更新処理を表している。

## 【0075】

最初に、ステップS151において、ライセンスサーバ4のCPU21は、クライアント1からのアクセスを受けると、ステップS152において、クライアント1がステップS135で送信したライセンス指定情報をライセンス更新要求情報とともに受信する。

## 【0076】

ステップS153において、CPU21は、ライセンスの更新要求を受信すると

、そのライセンスに対応する使用条件（更新する使用条件）を、記憶部28から読み出し、クライアント1に送信する。

## 【0077】

この提示に対して、上述したように、クライアント1から使用条件の購入が図10のステップS137の処理で申し込まれると、ステップS154において、ライセンスサーバ4のCPU21は、申し込まれた使用条件に対応するデータを生成し、ステップS154において、クライアントと1に送信する。クライアント1は、上述したように、ステップS139の処理で受信した使用条件を用いて、すでに登録されているライセンスの使用条件を更新する。

## 【0078】

本発明においては、図12に示されるように、ブロードキャストインクリプション（Broadcast Encryption）方式の原理に基づいて、デバイスとライセンスのキーが管理される。キーは、階層ツリー構造とされ、最下段のリーフ（leaf）が個々のデバイスまたはライセンスのキーに対応する。図12の例の場合、番号0から番号15までの16個のデバイスまたはライセンスに対応するキーが生成される。

## 【0079】

各キーは、図中丸印で示されるツリー構造の各ノードに対応して規定される。この例では、最上段のルートノードに対応してルートキーKRが、2段目のノードに対応してキーK0, K1が、3段目のノードに対応してキーK00乃至K11が、第4段目のノードに対応してキーK000乃至キーK111が、それぞれ対応されている。そして、最下段のノードとしてのリーフ（デバイスノード）に、キーK0000乃至K1111が、それぞれ対応されている。

## 【0080】

階層構造とされているため、例えば、キーK0010とキー0011の上位のキーは、K001とされ、キーK000とキーK001の上位のキーは、K00とされている。以下同様に、キーK00とキーK01の上位のキーは、K0とされ、キーK0とキーK1の上位のキーは、KRとされている。

## 【0081】

コンテンツを利用するキーは、最下段のデバイスノード（リーフ）から、最上段のルートノードまでの1つのパスの各ノードに対応するキーで管理される。例えば、番号3のノード（リーフID）に対応するライセンスに基づき、コンテンツを利用するキーは、キーK0011, K001, K00, K0, KRを含むパスの各キーで管理される。

#### 【0082】

本発明のシステムにおいては、図13に示されるように、図12の原理に基づいて構成されるMG-Rエンティティというキーシステムで、デバイスのキーとライセンスのキーの管理が行われる。図13の例では、8ラ24ラ32段のノードがツリー構造とされ、ルートノードから下位の8段までの各ノードにカテゴリが対応される。ここにおけるカテゴリとは、例えばメモリスティックなどの半導体メモリを使用する機器のカテゴリ、デジタル放送を受信する機器のカテゴリといったカテゴリを意味する。そして、このカテゴリノードのうちの1つのノードに、ライセンスを管理するシステムとしてのTシステムが対応される。

#### 【0083】

すなわち、このTシステムのノードよりさらに下の階層の24段のノードに対応するキーにより、ライセンスが対応される。この例の場合、これにより、約16メガ( $=2^{24}$ =約160万)のライセンスを規定することができる。さらに、最も下側の32段の階層により、約4ギガ( $=2^{32}$ =約40億)のユーザを規定することができる。最下段の32段のノードに対応するキーが、DNK(Device Node Key)を構成し、そのDNKに対応するIDがリーフIDとされる。

#### 【0084】

各デバイスやライセンスのキーは、64( $=8+24+32$ )段の各ノードで構成されるパスの内の1つに対応される。例えば、コンテンツを暗号化したコンテンツキーは、対応するライセンスに割り当てられたパスを構成するノードに対応するキーを用いて暗号化される。上位の階層のキーは、その直近の下位の階層のキーを用いて暗号化され、EKB(図15を参照して後述する)内に配置される。最下段のDNKは、EKB内には配置されず、ライセンスに記述され(図8)、ユーザのクライアント1に与えられる。クライアント1は、ライセンスに記

述されているDNKを用いて、コンテンツデータとともに配布されるEKB（図5）内に記述されている直近の上位の階層のキーを復号し、復号して得たキーを用いて、EKB内に記述されているさらにその上の階層のキーを復号する。以上の処理を順次行うことで、クライアント1は、そのコンテンツのパスに属するすべてのキーを得ることができる。

#### 【0085】

図14に階層ツリー構造のカテゴリーの分類の具体的な例を示す。図14において、階層ツリー構造の最上段には、ルートキーKR2301が設定され、以下の中間段にはノードキー2302が設定され、最下段には、リーフキー2303が設定される。各デバイスは個々のリーフキーと、リーフキーからルートキーに至る一連のノードキー、ルートキーを保有する。

#### 【0086】

最上段から第M段目（図13の例では、M=8）の所定のノードがカテゴリノード2304として設定される。すなわち第M段目のノードの各々が特定カテゴリのデバイス設定ノードとされる。第M段の1つのノードを頂点としてM+1段以下のノード、リーフは、そのカテゴリに含まれるデバイスに関するノードおよびリーフとされる。

#### 【0087】

例えば図14の第M段目の1つのノード2305にはカテゴリ【メモリステック（商標）】が設定され、このノード以下に連なるノード、リーフはメモリステックを使用した様々なデバイスを含むカテゴリ専用のノードまたはリーフとして設定される。すなわち、ノード2305以下が、メモリステックのカテゴリに定義されるデバイスの関連ノード、およびリーフの集合として定義される。

#### 【0088】

さらに、M段から数段分下位の段をサブカテゴリノード2306として設定することができる。図14の例では、カテゴリ【メモリステック】ノード2305の2段下のノードに、メモリステックを使用したデバイスのカテゴリに含まれるサブカテゴリノードとして、【再生専用器】のノード2306が設定されている。さらに、サブカテゴリノードである再生専用器のノード2306以下に、

再生専用器のカテゴリに含まれる音楽再生機能付き電話のノード2307が設定され、さらにその下位に、音楽再生機能付き電話のカテゴリに含まれる【P H S】ノード2308と、【携帯電話】ノード2309が設定されている。

#### 【0089】

さらに、カテゴリ、サブカテゴリは、デバイスの種類のみならず、例えはあるメーカー、コンテンツプロバイダ、決済機関等が独自に管理するノード、すなわち処理単位、管轄単位、あるいは提供サービス単位等、任意の単位（これらを総称して以下、エンティティと呼ぶ）で設定することが可能である。例えば1つのカテゴリノードをゲーム機器メーカーの販売するゲーム機器XYZ専用の頂点ノードとして設定すれば、メーカーの販売するゲーム機器XYZに、その頂点ノード以下の下段のノードキー、リーフキーを格納して販売することが可能となり、その後、暗号化コンテンツの配信、あるいは各種キーの配信、更新処理を、その頂点ノードキー以下のノードキー、リーフキーによって構成される有効化キーブロック（EKB）を生成して配信し、頂点ノード以下のデバイスに対してのみ利用可能なデータが配信可能となる。

#### 【0090】

このように、1つのノードを頂点として、以下のノードをその頂点ノードに定義されたカテゴリ、あるいはサブカテゴリの関連ノードとして設定する構成とすることにより、カテゴリ段、あるいはサブカテゴリ段の1つの頂点ノードを管理するメーカー、コンテンツプロバイダ等がそのノードを頂点とする有効化キーブロック（EKB）を独自に生成して、頂点ノード以下に属するデバイスに配信する構成が可能となり、頂点ノードに属さない他のカテゴリのノードに属するデバイスには全く影響を及ぼさずにキー更新を実行することができる。

#### 【0091】

例えば、図12に示されるツリー構造において、1つのグループに含まれる3つのデバイス0, 1, 2, 3はノードキーとして共通のキーK00, K0, KRを保有する。このノードキー共有構成を利用することにより、共通のコンテンツキーをデバイス0, 1, 2, 3のみに提供することが可能となる。たとえば、共通に保有するノードキーK00自体をコンテンツキーとして設定すれば、新たな鍵

送付を実行することなくデバイス0, 1, 2, 3のみが共通のコンテンツキーの設定が可能である。また、新たなコンテンツキーK<sub>con</sub>をノードキーK<sub>00</sub>で暗号化した値E<sub>nc</sub>(K<sub>00</sub>, K<sub>con</sub>)を、ネットワークを介してあるいは記録媒体に格納してデバイス0, 1, 2, 3に配布すれば、デバイス0, 1, 2, 3のみが、それぞれのデバイスにおいて保有する共有ノードキーK<sub>00</sub>を用いて暗号E<sub>nc</sub>(K<sub>00</sub>, K<sub>con</sub>)を解いてコンテンツキーK<sub>con</sub>を得ることが可能となる。なお、E<sub>nc</sub>(K<sub>a</sub>, K<sub>b</sub>)はK<sub>b</sub>をK<sub>a</sub>によって暗号化したデータであることを示す。

#### 【0092】

また、ある時点tにおいて、デバイス3の所有する鍵K<sub>0011</sub>, K<sub>001</sub>, K<sub>00</sub>, K<sub>0</sub>, KRが攻撃者（ハッカー）により解析されて露呈したことが発覚した場合、それ以降、システム（デバイス0, 1, 2, 3のグループ）で送受信されるデータを守るために、デバイス3をシステムから切り離す必要がある。そのためには、ノードキーK<sub>001</sub>, K<sub>00</sub>, K<sub>0</sub>, KRを、それぞれ新たな鍵K(t)001, K(t)00, K(t)0, K(t)Rに更新し、デバイス0, 1, 2にその更新キーを伝える必要がある。ここで、K(t)aaaは、鍵Kaaaの世代（Generation）tの更新キーであることを示す。

#### 【0093】

更新キーの配布処理について説明する。キーの更新は、例えば、図15Aに示す有効化キーブロック（EKB : Enabling Key Block）と呼ばれるブロックデータによって構成されるテーブルを、ネットワークを介して、あるいは記録媒体に格納してデバイス0, 1, 2に供給することによって実行される。なお、有効化キーブロック（EKB）は、図12に示されるようなツリー構造を構成する各リーフ（最下段のノード）に対応するデバイスに、新たに更新されたキーを配布するための暗号化キーによって構成される。有効化キーブロック（EKB）は、キー更新ブロック（KRБ : Key Renewal Block）と呼ばれることもある。

#### 【0094】

図15Aに示す有効化キーブロック（EKB）は、ノードキーの更新の必要なデバイスのみが更新可能なデータ構成を持つブロックデータとして構成される。

図15Aの例は、図12に示すツリー構造中のデバイス0, 1, 2において、世代tの更新ノードキーを配布することを目的として形成されたブロックデータである。図12から明らかなように、デバイス0, デバイス1は、更新ノードキーとして $K(t)00$ 、 $K(t)0$ 、 $K(t)R$ が必要であり、デバイス2は、更新ノードキーとして $K(t)001$ 、 $K(t)00$ 、 $K(t)0$ 、 $K(t)R$ が必要である。

#### 【0095】

図15AのEKBに示されるように、EKBには複数の暗号化キーが含まれる。図15Aの最下段の暗号化キーは、 $Enc(K0010, K(t)001)$ である。これはデバイス2の持つリーフキー $K0010$ によって暗号化された更新ノードキー $K(t)001$ であり、デバイス2は、自身の持つリーフキー $K0010$ によってこの暗号化キーを復号し、更新ノードキー $K(t)001$ を得ることができる。また、復号により得た更新ノードキー $K(t)001$ を用いて、図15Aの下から2段目の暗号化キー $Enc(K(t)001, K(t)00)$ が復号可能となり、更新ノードキー $K(t)00$ を得ることができる。

#### 【0096】

以下順次、図15Aの上から2段目の暗号化キー $Enc(K(t)00, K(t)0)$ を復号することで、更新ノードキー $K(t)0$ が得られ、これを用いて、図15Aの上から1段目の暗号化キー $Enc(K(t)0, K(t)R)$ を復号することで、更新ルートキー $K(t)R$ が得られる。

#### 【0097】

一方、ノードキー $K000$ は更新する対象に含まれておらず、ノード0, 1が、更新ノードキーとして必要なのは、 $K(t)00$ 、 $K(t)0$ 、 $K(t)R$ である。ノード0, 1は、デバイスキーア $K0000$ ,  $K0001$ を用いて、図15Aの上から3段目の暗号化キー $Enc(K000, K(t)00)$ を復号することで更新ノードキー $K(t)00$ を取得し、以下順次、図15Aの上から2段目の暗号化キー $Enc(K(t)00, K(t)0)$ を復号することで、更新ノードキー $K(t)0$ を得、図15Aの上から1段目の暗号化キー $Enc(K(t)0, K(t)R)$ を復号することで、更新ルートキー $K(t)R$ を得る。このよ

うにして、デバイス0, 1, 2は更新したキーK(t)Rを得ることができる。

#### 【0098】

なお、図15Aのインデックスは、図の右側の暗号化キーを復号するための復号キーとして使用するノードキー、リーフキーの絶対番地を示す。

#### 【0099】

図12に示すツリー構造の上位段のノードキーK(t)0,K(t)Rの更新が不要であり、ノードキーK00のみの更新処理が必要である場合には、図15Bの有効化キーブロック(EKB)を用いることで、更新ノードキーK(t)00をデバイス0, 1, 2に配布することができる。

#### 【0100】

図15Bに示すEKBは、例えば特定のグループにおいて共有する新たなコンテンツキーを配布する場合に利用可能である。具体例として、図12に点線で示すグループ内のデバイス0, 1, 2, 3がある記録媒体を用いており、新たな共通のコンテンツキーK(t)conが必要であるとする。このとき、デバイス0, 1, 2, 3の共通のノードキーK00を更新したK(t)00を用いて新たな共通の更新コンテンツキーK(t)conを暗号化したデータEnc(K(t)00, K(t)con)が、図15Bに示されるEKBとともに配布される。この配布により、デバイス4など、その他のグループの機器が復号することができないデータとしての配布が可能となる。

#### 【0101】

すなわち、デバイス0, 1, 2はEKBを処理して得たキーK(t)00を用いて暗号文を復号すれば、t時点でのコンテンツキーK(t)conを得ることが可能になる。

#### 【0102】

図16に、t時点でのコンテンツキーK(t)conを得る処理例として、K(t)00を用いて新たな共通のコンテンツキーK(t)conを暗号化したデータEnc(K(t)00, K(t)con)と、図15Bに示すEKBとを記録媒体を介して受領したデバイス0の処理を示す。すなわちこの例は、EKBによる暗号化メッセージデータをコンテンツキーK(t)conとした例である。

## 【0103】

図16に示すように、デバイス0は、記録媒体に格納されている世代t時点のEKBと、自分があらかじめ格納しているノードキーK000を用いて、上述したと同様のEKB処理により、ノードキーK(t)00を生成する。さらに、デバイス0は、復号した更新ノードキーK(t)00を用いて、更新コンテンツキーK(t)conを復号して、後にそれを使用するために自分だけが持つリーフキーK0000で暗号化して格納する。

## 【0104】

図17に有効化キーブロック(EKB)のフォーマット例を示す。バージョン601は、有効化キーブロック(EKB)のバージョンを示す識別子である。なお、バージョンは、最新のEKBを識別する機能と、コンテンツとの対応関係を示す機能を持つ。デプスは、有効化キーブロック(EKB)の配布先のデバイスに対する階層ツリーの階層数を示す。データポインタ603は、有効化キーブロック(EKB)中のデータ部606の位置を示すポインタであり、タグポインタ604はタグ部607の位置、署名ポインタ605は署名608の位置を示すポインタである。

## 【0105】

データ部606は、例えば更新するノードキーを暗号化したデータを格納する。例えば図16に示すような更新されたノードキーに関する各暗号化キー等を格納する。

## 【0106】

タグ部607は、データ部606に格納された暗号化されたノードキー、リーフキーの位置関係を示すタグである。このタグの付与ルールを図18を用いて説明する。

## 【0107】

図18では、データとして先に図15Aで説明した有効化キーブロック(EKB)を送付する例を示している。この時のデータは、図18Bの表に示すようになる。このときの暗号化キーに含まれるトップノードのアドレスをトップノードアドレスとする。この例の場合は、ルートキーの更新キーK(t)Rが含まれて

いるので、トップノードアドレスはKRとなる。このとき、例えば最上段のデータ  $E_{nc}(K(t)0, K(t)R)$  は、図18Aに示す階層ツリーに示す位置P0に対応する。次の段のデータは、 $E_{nc}(K(t)00, K(t)0)$  であり、ツリー上では前のデータの左下の位置P00に対応する。ツリー構造の所定の位置から見て、その下に、データがある場合は、タグが0、ない場合はタグが1に設定される。タグは {左(L)タグ, 右(R)タグ} として設定される。図18Bの最上段のデータ  $E_{nc}(K(t)0, K(t)R)$  に対応する位置P0の左下の位置P00にはデータがあるので、Lタグ=0、右にはデータがないので、Rタグ=1となる。以下、すべてのデータにタグが設定され、図18Cに示すデータ列、およびタグ列が構成される。

#### 【0108】

タグは、対応するデータ  $E_{nc}(Kxxx, Kyyy)$  が、ツリー構造のどこに位置しているのかを示すために設定されるものである。データ部606に格納されるキーデータ  $E_{nc}(Kxxx, Kyyy) \dots$  は、単純に暗号化されたキーの羅列データに過ぎないが、上述したタグによってデータとして格納された暗号化キーのツリー上の位置が判別可能となる。上述したタグを用いずに、先の図15で説明した構成のように、暗号化データに対応させたノード・インデックスを用いで、例えば、

0 :  $E_{nc}(K(t)0, K(t)R)$   
 00 :  $E_{nc}(K(t)00, K(t)0)$   
 000 :  $E_{nc}(K((t)000, K(t)00)$

…のようなデータ構成とすることも可能であるが、このようなインデックスを用いた構成とすると、冗長なデータとなりデータ量が増大し、ネットワークを介する配信等においては好ましくない。これに対し、上述したタグをキー位置を示す索引データとして用いることにより、少ないデータ量でキー位置の判別が可能となる。

#### 【0109】

図17に戻って、EKBフォーマットについてさらに説明する。署名(Signature) 608は、有効化キーブロック(EKB)を発行した例えば鍵管理センタ

(ライセンスサーバ4)、コンテンツロバイダ(コンテンツサーバ3)、決済機関(課金サーバ5)等が実行する電子署名である。EKBを受領したデバイスは、署名検証によって正当な有効化キーブロック(EKB)発行者が発行した有効化キーブロック(EKB)であることを確認する。

#### 【0110】

以上のようにして、ライセンスサーバ4から供給されたライセンスに基づいて、コンテンツサーバ3から供給されたコンテンツを利用する処理をまとめると、図19に示されるようになる。

#### 【0111】

すなわち、コンテンツサーバ3からクライアント1に対してコンテンツが提供されるとともに、ライセンスサーバ4からクライアント1にライセンスが供給される。ライセンスには、DNKが含まれている(図8)。コンテンツは、コンテンツキーKcにより、暗号化されており(Enc(Kc, Content))、コンテンツキーKcは、ルートキーKR(EKBから得られるキーであって、図5におけるキーK<sub>EKBC</sub>に対応する)で暗号化され(Enc(KR, Kc))、EKBとともに、暗号化されたコンテンツに付加されてクライアント1に提供される。

#### 【0112】

図19の例におけるEKBには、例えば、図20に示されるように、DNKで暗号化したルートキーKRが含まれている(Enc(DNK, KR))。従って、クライアント1は、ライセンスに含まれるDNKを利用して、EKBからルートキーKRを得ることができる。さらに、ルートキーKRを用いて、Enc(KR, Kc)からコンテンツキーKcを復号することができ、コンテンツキーKcを用いて、Enc(Kc, Content)からコンテンツを復号することができる。

#### 【0113】

このように、ライセンスにDNKを個別に割り当てることにより、図1.2と図1.5を参照して説明した原理に従って、個々のライセンスのリボーカ(revoke)が可能になる。

#### 【0114】

また、ライセンスをユーザ毎に配布することにより、ライセンスサーバ4とク

ライアント1において、リーフIDとDNKの対応付けが行われることになり、ライセンスの不正コピーを防止することが可能になる。

#### 【0115】

また、ライセンスをキー(DNKとリーフID)と使用条件の2つから構成することにより(図8)、再暗号化等をすることなく、使用条件を変更することが可能になる。また、証明書と秘密鍵をライセンスとともに配信するようになると、エンドユーザも、これらを用いて不正コピーを防止可能なコンテンツを作成することが可能になる。

#### 【0116】

証明書と秘密鍵の利用については、図28のフローチャートを参照して後述する。

#### 【0117】

本発明においては、図13を参照して説明したように、カテゴリノードにライセンスを管理するTシステムと、各種のコンテンツを利用するデバイスのカテゴリが対応づけられるので、複数のDNKを1つの(共通の)デバイスに持たせることができる。その結果、異なるカテゴリのコンテンツを1つのデバイスで管理することが可能となる。

#### 【0118】

図21は、この関係を表している。すなわち、デバイスD1には、Tシステムに基づいて、DNK1が割り当てられている、コンテンツ1を利用するライセンスが記録される。同様に、このデバイスD1には、例えば、DNK2が割り当てられた、メモリスティックにCDからリッピングしたコンテンツ2を記録することができる。この場合、デバイスD1は、コンテンツ1とコンテンツ2という、異なるシステム(Tシステムとデバイス管理システム)により配信されたコンテンツを同時に扱うことが可能となる。新たなDNKを割り当てるとき、既に割り当てられているDNKを削除するなどして、デバイスに1個のDNKだけを対応させるようにした場合、このようなことはできない。

#### 【0119】

また、図13における、例えば、下側の32階層の各三角形の1つ1つに、図

22に示されるライセンスカテゴリ1とライセンスカテゴリ2を割り当てることにより、同一のカテゴリ内を、サブカテゴリを利用して、コンテンツのジャンル、レベル、販売店、配信サービス等の小さな集まりに分類して、管理することが可能となる。

#### 【0120】

図22の例においては、例えば、ライセンスカテゴリ1は、ジャズのジャンルに属し、ライセンスカテゴリ2は、ロックのジャンルに属する。ライセンスカテゴリ1には、ライセンスIDが1であるコンテンツ1とコンテンツ2を対応させ、それぞれユーザ1乃至ユーザ3に配布されている。ライセンスカテゴリ2は、ライセンスID2のコンテンツ3、コンテンツ4、およびコンテンツ5が含まれ、それぞれユーザ1とユーザ3に提供されている。

#### 【0121】

このように、本発明においては、カテゴリ毎に独立したキー管理が可能になる。

#### 【0122】

また、DNKを、機器やメディアに予め埋め込むのではなく、ライセンスサーバ4により、その場で生成し、各機器やメディアにダウンロードするようになると、ユーザによるキーの購入が可能なシステムを実現することができる。

#### 【0123】

コンテンツは、それが作成された後、どのような使われ方をされようとも、その使われ方に関わりなく、全ての用途において、使用可能とされるべきである。例えば、異なるコンテンツ配信サービス、あるいは使用条件が異なるドメイン等でも、同一のコンテンツが使えることが望ましい。本発明においては、このため、上述したように、各ユーザ（クライアント1）に、認証局としてのライセンスサーバ4から秘密鍵と、それに対応する公開鍵の証明書（certificates）が配布される。各ユーザは、その秘密鍵を用いて、署名（signature）を作成し、コンテンツに付加して、コンテンツの真正さを保証し、かつコンテンツの改竄防止を図ることができる。

#### 【0124】

この場合の処理の例について、図23のフローチャートを参照して説明する。図23の処理は、ユーザがCDから再生したデータを記憶部28に記憶させるリッピング処理を説明するものである。

#### 【0125】

最初に、ステップS171において、クライアント1のCPU21は、通信部29を介して入力されるCDの再生データを記録データとして取り込む。ステップS172において、CPU21は、ステップS171の処理で取り込まれた記録データにウォーターマークが含まれているか否かを判定する。このウォーターマークは、3ビットのコピー管理情報(CCI)と、1ビットのトリガ(Trigger)により構成され、コンテンツのデータの中に埋め込まれているCPU21は、ウォーターマークが検出された場合には、ステップS173に進み、そのウォーターマークを抽出する処理を実行する。ウォーターマークが存在しない場合には、ステップS173の処理はスキップされる。

#### 【0126】

次に、ステップS174において、CPU21は、コンテンツに対応して記録するヘッダのデータを作成する。このヘッダのデータは、コンテンツID、ライセンスID、ライセンスを取得するためのアクセス先を表すURL、およびウォーターマークにより構成される。

#### 【0127】

次に、ステップS175に進み、CPU21は、ステップS174の処理で作成したヘッダのデータに基づいたデジタル署名を、自分自身の秘密鍵を用いて作成する。この秘密鍵は、ライセンスサーバ4から取得したものである(図7のステップS67)。

#### 【0128】

ステップS176で、CPU21は、暗号化復号部24を制御し、コンテンツキーでコンテンツを暗号化させる。コンテンツキーは、コンテンツを取得したとき、同時に取得されたものである(図5または図19)。

#### 【0129】

次に、ステップS177において、CPU21は、ファイルフォーマットに基づ

き、データを、例えば、ミニディスク等により構成される光磁気ディスク43に記録させる。

#### 【0130】

なお、記録媒体がミニディスクである場合、ステップS176において、CPU21は、コンテンツをコーデック部25に供給し、例えば、ATRAC3方式によりコンテンツを符号化させる。そして、符号化されたデータが符号化復号部24によりさらに暗号化される。

#### 【0131】

図24は、以上のようにして、記録媒体にコンテンツが記録された状態を模式的に表している。暗号化されているコンテンツ（E（At3））から抽出されたウォーターマーク（WM）が、コンテンツの外（ヘッダ）に記録されている。

#### 【0132】

図25は、コンテンツを記録媒体に記録する場合のファイルフォーマットのより詳細な構成を表している。この例においては、コンテンツID（CID）、ライセンスID（LID）、URL、およびウォーターマーク（WM）を含むヘッダが記録されている他、EKB、コンテンツキーKcをルートキーKRで暗号化したデータ（Enc（KR, Kc））、証明書（Cert）、ヘッダに基づき生成されたデジタル署名（Sig（Header））、コンテンツをコンテンツキーKcで暗号化したデータ（Enc（Kc, Content））、メタデータ（Meta Data）およびマーク（Mark）が記録されている。

#### 【0133】

ウォーターマークは、コンテンツの内部に埋め込まれているものであるが、図24と図25に示されるように、コンテンツの内部とは別に、ヘッダ内に配置するようにすることで、ウォーターマークを迅速に、かつ簡単に検出することが可能となる。従って、そのコンテンツを、コピーすることができるか否かを、迅速に判定することができる。

#### 【0134】

なお、メタデータは、例えば、ジャケット、写真、歌詞等のデータを表す。マークについては、図31を参照して後述する。

#### 【0135】

図26は、証明書としての公開鍵証明書の例を表している。公開鍵証明書は、通常、公開鍵暗号方式における認証局（CA: Certificate Authority）が発行する証明書であり、ユーザが、認証局に提出した自己のIDや公開鍵などに、認証局が有効期限等の情報を付加し、さらに、認証局によるデジタル署名を付加して作成される。この発明においては、ライセンスサーバ4（またはコンテンツサーバ3）が、証明書と秘密鍵、従って公開鍵も発行するので、ユーザは、IDとパスワードをライセンスサーバ4に提供するだけで、この公開鍵証明書を得ることができる（図7のステップS67）。

#### 【0136】

図26における公開鍵証明書は、証明書のバージョン番号、ライセンスサーバ4が証明書の利用者（ユーザ）に対して割りつける証明書の通し番号、デジタル署名に用いたアルゴリズム、およびパラメータ、認証局（ライセンスサーバ4）の名前、証明書の有効期限、証明書利用者のID（ノードIDまたはリーフID）、並びに証明書利用者の公開鍵が、メッセージとして含まれている。さらに、このメッセージには、認証局としてのライセンスサーバ4により作成されたデジタル署名が付加されている。このデジタル署名は、メッセージに対してハッシュ関数を適用を提供して生成されたハッシュ値に基づいて、ライセンスサーバ4の秘密鍵を用いて生成されたデータである。

#### 【0137】

ノードIDまたはリーフIDは、例えば、図12の例の場合、デバイス0であれば「0000」とされ、デバイス1であれば「0001」とされ、デバイス15であれば「1111」とされる。このようなIDに基づいて、そのデバイス（エンティティ）がツリー構成のどの位置（リーフまたはノード）に位置するエンティティであるのかが識別される。

#### 【0138】

このように、コンテンツを利用するのに必要なライセンスを、コンテンツとは分離して配布することにより、コンテンツの配布が自由に行われることになる。任意の方法、あるいは経路で入手されたコンテンツは、一元的に取り扱うことが可能である。

## 【0139】

また、ファイルフォーマットを図25に示されるように構成することで、そのフォーマットのコンテンツを、インターネットを介して配信する場合は勿論、SDMI (Secure Digital Music Initiative) 機器に提供する場合においても、コンテンツの著作権を管理することが可能となる。

## 【0140】

さらに、例えば、図27に示されるように、コンテンツが記録媒体を介して提供されたとしても、インターネット2を介して提供されたとしても、同様の処理により、SDMI (Secure Digital Music Initiative) 機器としての所定のPD (Portable Device) 等に、チェックアウトしたりすることが可能となる。

## 【0141】

次に、図28のフローチャートを参照して、クライアント1が他のクライアント（例えば、PD）に対してコンテンツをチェックアウトする場合の処理について説明する。

## 【0142】

最初に、ステップS191において、CPU21は、コンテンツにデジタル署名が付加されているか否かを判定する。デジタル署名が付加されていると判定された場合、ステップS192に進み、CPU21は、証明書を抽出し、認証局（ライセンスサーバ4）の公開鍵で検証する処理を実行する。すなわち、クライアント1は、ライセンスサーバ4からライセンスサーバ4の秘密鍵に対応する公開鍵を取得し、その公開鍵で公開鍵証明書に付加されているデジタル署名を復号する。図26を参照して説明したように、デジタル署名は、認証局（ライセンスサーバ4）の秘密鍵に基づいて生成されており、ライセンスサーバ4の公開鍵を用いて復号することができる。さらに、CPU21は、証明書のメッセージ全体に対してハッシュ関数を適用してハッシュ値を演算する。そしてCPU21は、演算されたハッシュ値と、デジタル署名を復号して得られたハッシュ値とを比較し、両者が一致すれば、メッセージは改竄されたものではないと判定する。両者が一致しない場合には、この証明書は、改竄されたものであるということになる。

## 【0143】

そこで、ステップS193において、CPU21は、証明書が改竄されていないか否かを判定し、改竄されていないと判定された場合、ステップS194に進み、証明書をEKBで検証する処理を実行する。この検証処理は、証明書に含まれるリーフID（図26）に基づいて、EKBをたどることができるか否かを調べることにより行われる。この検証について、図29と図30を参照して説明する。

#### 【0144】

いま、図29に示されるように、例えば、リーフキーK1001を有するデバイスがリボーカされたデバイスであるとする。このとき、図30に示されるようなデータ（暗号化キー）とタグを有するEKBが、各デバイス（リーフ）に配布される。このEKBは、図29におけるデバイス「1001」をリボーカするために、キーKR,K1,K10,K100を更新するEKBとなっている。

#### 【0145】

リボーカデバイス「1001」以外の全てのリーフは、更新されたルートキーK(t)Rを取得することができる。すなわち、ノードキーK0の下位に連なるリーフは、更新されていないノードキーK0を、デバイス内に保持しているので、暗号化キーEnc(K0, K(t)R)を、キーK0によって復号することで、更新ルートキーK(t)Rを取得することができる。

#### 【0146】

また、ノードキーK11以下のリーフは、更新されていないノードキーK11を用いて、Enc(K11, K(t)1)をノードキーK11によって復号することで、更新ノードキーK(t)1を取得することができる。さらに、Enc(K(t)1, K(t)R)をノードキーK(t)1によって復号することで、更新ルートキーK(t)Rを取得することが可能となる。ノードキーK101の下位リーフについても、同様に更新ルートキーK(t)Rを取得することが可能である。

#### 【0147】

さらに、リボーカされていないリーフキーK1000を有するデバイス「1000」は、自己のリーフキーK1000でEnc(K1000, K(t)100)を復号して、ノードキーK(t)100を取得することができ、これを用いてさ

らに、上位のノードキーを順次復号し、更新ルートキーK(t)Rを取得することができる。

#### 【0148】

これに対して、リボーカされたデバイス「1001」は、自己のリーフの1段上の更新ノードキーK(t)100を、EKB処理により取得できないので、結局、更新ルートキーK(t)Rを取得することができない。

#### 【0149】

リボーカされていない正当なデバイス（クライアント1）には、図30に示されるデータとタグを有するEKBが、ライセンスサーバ4から配信され、格納されている。

#### 【0150】

そこで、各クライアントは、そのタグを利用して、EKB追跡処理を行うことができる。このEKB追跡処理は、上位のルートキーからキー配信ツリーをたどれるか否かを判定する処理である。

#### 【0151】

例えば、図29のリーフ「1001」のID（リーフID）である「1001」を、「1」「0」「0」「1」の4ビットとして把握し、最上位ビットから順次、下位ビットに従って、ツリーをたどることができるか否かが判定される。この判定では、ビットが1であれば、右側に進み、0であれば、左側に進む処理が行われる。

#### 【0152】

ID「1001」の最上位ビットが1であるから、図29のルートキーKRから右側に進む。EKBの最初のタグ（番号0のタグ）は、0：{0, 0}であり、両枝にデータを有するものであると判定される。この場合、右側に進むことができるので、ノードキーK1にたどり着くことができる。

#### 【0153】

次に、ノードキーK1の下位のノードに進む。ID「1001」の2番目のビットは0であるから左側に進む。番号1のタグは、左側のノードキーK0の下位のデータの有無を表すものであり、ノードキーK1の下位のデータの有無を示すタ

グは、番号2のタグである。このタグは、図30に示されるように、 $2 : \{0, 0\}$ であり、両枝にデータを有するものとされる。従って、左側に進み、ノードキーK10にたどり着くことができる。

#### 【0154】

さらに、ID「1001」の3番目のビットは0であり、左側に進む。このとき、K10の下位のデータの有無を示すタグ（番号3のタグ）は、 $3 : \{0, 0\}$ であり、両枝にデータを有するものと判定される。そこで、左側に進み、ノードキーK100にたどり着くことができる。

#### 【0155】

さらに、ID「1001」の最下位ビットは1であり、右側に進む。番号4のタグは、ノードキーK11に対応するものであり、K100の下位のデータの符号を表すタグは、番号5のタグである。このタグは、 $5 : \{0, 1\}$ である。従って、右側には、データが存在しないことになる。その結果、ノード「1001」にはたどり着けないことになり、ID「1001」のデバイスは、EKBによる更新ルートキーを取得できないデバイス、すなわちリボーケデバイスであると判定される。

#### 【0156】

これに対して、例えば、リーフキーK1000を有するデバイスIDは、「1000」であり、上述した場合と同様に、EKB内のタグに基づくEKB追跡処理を行うと、ノード「1000」にたどり着くことができる。従って、ID「1000」のデバイスは、正当なデバイスであると判定される。

#### 【0157】

図28に戻って、CPU21は、ステップS194の検証処理に基づき、証明書が改竄されていないか否かをステップS195で判定し、証明書が改竄されていない場合には、ステップS196に進み、デジタル署名を証明書に含まれる公開鍵で検証する処理を実行する。

#### 【0158】

すなわち、図26に示されるように、証明書には、証明書利用者（コンテンツ作成者）の公開鍵が含まれており、この公開鍵を用いて、図25に示される署名

(Sig (Header) ) が検証される。すなわち、この公開鍵を用いて、デジタル署名Sig (Header) を復号して得られたデータ（ハッシュ値）と、図25に示されるHeaderにハッシュ関数を適用して演算されたハッシュ値とを比較することで、両者が一致していれば、Headerが改竄されていないことを確認することができる。これに対して、両者が一致しなければ、Headerは改竄されているということになる。

## 【0159】

ステップS197において、CPU21は、Headerが改竄されているか否かを判定し、改竄されていなければ、ステップS198に進み、ウォーターマークを検証する。ステップS199において、CPU21は、ウォーターマークの検証の結果、チェックアウトが可能であるか否かを判定する。チェックアウトが可能である場合には、ステップS200に進み、CPU21は、チェックアウトを実行する。すなわち、チェックアウト先のクライアント1に対してコンテンツを転送し、コピーさせる。

## 【0160】

ステップS191において、デジタル署名が存在しないと判定された場合、ステップS193において、証明書が改竄されていると判定された場合、ステップS195において、証明書をEKBで検証することができなかつたと判定された場合、ステップS197において、デジタル署名の検証の結果、ヘッダが改竄されていると判定された場合、または、ステップS199において、ウォーターマークにチェックアウトの禁止が記述されていると判定された場合、ステップS201に進み、エラー処理が実行される。すなわち、この場合には、チェックアウトが禁止される。

## 【0161】

このように、証明書と秘密鍵をライセンスサーバ4からユーザに配布し、コンテンツ作成時に、デジタル署名を付加することにより、コンテンツの作成者の真正を保証することが可能となる。これにより、不正なコンテンツの流通を抑制することができる。

## 【0162】

さらに、ウォーターマークをコンテンツ作成時に検出し、その情報をデジタル署名に付することで、ウォーターマーク情報の改竄を防止し、コンテンツの真正を保証することができる。

#### 【0163】

その結果、一度作成されたコンテンツは、どのような形態で配信されたとしても、元のコンテンツの真正を保証することが可能となる。

#### 【0164】

さらに、コンテンツは、使用条件を有さず、使用条件は、ライセンスに付加されているので、ライセンス内の使用条件を変更することで、それに関係するコンテンツの使用条件を一斉に変更することが可能となる。

#### 【0165】

次に、マークの利用方法について説明する。本発明においては、上述したように、使用条件は、コンテンツではなく、ライセンスに付加される。しかしながら、コンテンツによって、使用状況が異なる場合がある。そこで、本発明においては、図25に示されるように、コンテンツにマークが付加される。

#### 【0166】

ライセンスとコンテンツは、1対多の関係にあるため、コンテンツの個々の使用状況をライセンスの使用条件にのみ記述するのは困難となる。そこで、このように、コンテンツに使用状況を付加することにより、ライセンスでの管理をしながらも、個々のコンテンツを管理することが可能となる。

#### 【0167】

このマークには、例えば、図31に示されるように、ユーザのID(リーフID)、所有権フラグ、使用開始時刻、およびコピー回数等が記述される。

#### 【0168】

さらに、マークには、リーフID、所有権フラグ、使用開始時刻、およびコピー回数等のメッセージに基づいて生成されたデジタル署名が付加される。

#### 【0169】

所有権フラグは、例えば、所定の期間だけコンテンツを使用可能とするライセンスを、そのまま買い取ったような場合(使用期間を永久に変更したような場合

) に付加される。使用開始時刻は、コンテンツの使用を所定の期間内に開始した場合に記述される。例えば、コンテンツをダウンロードする時期が制限されているような場合において、その期限内にダウンロードが行われたようなとき、その実際にコンテンツをダウンロードした日時がここに記述される。これにより、期間内での有効な使用であることが、証明される。

## 【0170】

コピー回数には、それまでにそのコンテンツをコピーした回数が履歴（ログ）として記述される。

## 【0171】

次に、図32のフローチャートを参照して、ユーザがライセンスを買い取った場合に、マークを付加する処理について、マークをコンテンツに付加する例として説明する。

## 【0172】

最初に、ステップS221において、CPU21は、入力部26からのユーザの指令に基づいて、インターネット2を介して、ライセンスサーバ4にアクセスする。

## 【0173】

ステップS22.2において、CPU21は、ユーザからの入力部26を介しての入力を取り込み、その入力に対応してライセンスサーバ4に対してライセンスの買い取りを要求する。

## 【0174】

この要求に対応して、図33のフローチャートを参照して後述するように、ライセンスサーバ4は、ライセンスを買い取るために必要な対価を提示していく（図33のステップS242）。そこで、ステップS223において、クライアント1のCPU21は、ライセンスサーバ4からの対価の提示を受け取ると、これを出力部27に出力し、表示させる。

## 【0175】

ユーザは、この表示に基づいて、提示された対価を了承するか否かを判断し、その判断結果に基づいて、入力部26からその判断結果を入力する。

## 【0176】

CPU2 1は、ステップS224において、入力部26からの入力に基づいて、ユーザが提示された対価を了承したか否かを判定し、了承したと判定した場合には、ステップS225に進み、ライセンスサーバ4に了承を通知する処理を実行する。

## 【0177】

この了承通知を受信すると、ライセンスサーバ4は、対価の買い取りを表す情報、すなわち所有権フラグを記述したマークを送信してくる（図33のステップS244）。そこで、ステップS226において、クライアント1のCPU2 1は、ライセンスサーバ4からのマークを受け取ると、ステップS277において、受け取ったマークをコンテンツに埋め込む処理を実行する。すなわち、これにより、買い取られたライセンスに対応するコンテンツのマークとして、図31に示されるような所有権フラグが記述されたマークがコンテンツに対応して記録されることになる。また、このとき、CPU2 1は、メッセージが更新されたことになるので、デジタル署名（図25）も更新し、記録媒体に記録する。

## 【0178】

ステップS224において、ライセンスサーバ4から提示された対価が了承されていないと判定された場合、ステップS228に進み、CPU2 1は、提示された対価を了承しないことをライセンスサーバ4に通知する。

## 【0179】

このようなクライアント1の処理に対応して、ライセンスサーバ4は、図33のフローチャートに示す処理を実行する。

## 【0180】

すなわち、最初に、ステップS241において、ライセンスサーバ4のCPU2 1は、クライアント1からライセンス買い取りの要求が送信されると（図30のステップS222）、これを受け取り、ステップS242において、対象とされているライセンスの買い取りに必要な対価を記憶部28から読み出し、これをクライアント1に送信する。

## 【0181】

上述したように、このようにして提示された対価に対して、クライアント1から提示された対価を了承するか否かの通知が送信されてくる。

#### 【0182】

そこで、ステップS243において、ライセンスサーバ4のCPU21は、クライアント1から了承通知を受信したか否かを判定し、了承通知を受信したと判定した場合、ステップS244に進み、対象とされるライセンスの買い取りを表すメッセージを含むマークを生成し、自分自身の秘密鍵で、デジタル署名を附加して、クライアント1に送信する。このようにして送信されたマークは、上述したように、クライアント1の記憶部28において、対応するコンテンツに記録される（図32のステップS227）。

#### 【0183】

ステップS243において、クライアント1から了承通知が受信されていないと判定された場合、ステップS244の処理はスキップされる。すなわち、この場合には、ライセンスの買い取り処理が最終的に行われなかることになるので、マークは送信されない。

#### 【0184】

図34は、ステップS244において、ライセンスサーバ4からクライアント1に対して送信されるマークの構成例を表している。この例においては、そのユーザのリーフID、所有権フラグ（Own）、並びにリーフIDと所有権フラグを、ライセンスサーバ4の秘密鍵Sに基づいて生成されたデジタル署名Sig<sub>S</sub>（LeafID, Own）により、マークが構成されている。

#### 【0185】

なお、このマークは、特定のユーザの特定のコンテンツに対してのみ有効なものであるので、対象とされるコンテンツがコピーされた場合には、そのコピーされたコンテンツに付随するマークは無効とされる。

#### 【0186】

このようにして、コンテンツとライセンスを分離し、使用条件をライセンスに対応させる場合においても、個々のコンテンツの使用状況に応じたサービスを実現することが可能となる。

## 【0187】

次に、グルーピングについて説明する。複数の機器やメディアを適当に集め、その1つの集合内においては、コンテンツを自由に授受することができるようすることは、グルーピングと称される。通常、このグルーピングは、個人の所有する機器やメディアにおいて行われる。このグルーピングは、従来、グループ毎にグループキーを設定する等して行われていたが、グループ化する複数の機器やメディアに、同一のライセンスを対応づけることにより、容易にグルーピングすることが可能となる。

## 【0188】

また、各機器を予め登録しておくことで、グルーピングすることも可能である。この場合のグルーピングについて、以下に説明する。

## 【0189】

この場合、ユーザは、グルーピング対象とされる機器の証明書を予めサーバに登録しておく必要がある。この証明書の登録処理について、図35と図36のフローチャートを参照して説明する。

## 【0190】

最初に、図35のフローチャートを参照して、クライアント（グルーピング対象となる機器）の証明書の登録処理について説明する。ステップS261において、クライアント1のCPU21は、グルーピングの対象とされる機器としての自分自身の証明書を作成する。この証明書には、自分自身の公開鍵が含まれる。

## 【0191】

次に、ステップS262に進み、CPU21は、ユーザの入力部26からの入力に基づいて、コンテンツサーバ3にアクセスし、ステップS263において、ステップS261の処理で作成された証明書をコンテンツサーバ3に送信する処理を実行する。

## 【0192】

なお、証明書としては、ライセンスサーバ4から受信したものを、そのまま使用することもできる。

## 【0193】

以上の処理は、グルーピング対象とされる全ての機器が行う。

#### 【0194】

次に、図36のフローチャートを参照して、図35のクライアント1の証明書の登録処理に対応して行われるコンテンツサーバ3の証明書の登録処理について説明する。

#### 【0195】

最初に、ステップS271において、コンテンツサーバ3のCPU21は、クライアント1から送信されてきた証明書を受信すると、ステップS272において、その証明書を記憶部28に登録する。

#### 【0196】

以上の処理が、グループ対象とされる機器毎に行われる。その結果、コンテンツサーバ3の記憶部28には、例えば、図37に示されるように、グループ毎に、そのグループを構成するデバイスの証明書が登録される。

#### 【0197】

図37に示される例では、グループ1の証明書として、証明書C11乃至C14が登録されている。これらの証明書C11乃至C14には、対応する公開鍵K<sub>P11</sub>乃至K<sub>P14</sub>が含まれている。

#### 【0198】

同様に、グループ2の証明書として、証明書C21乃至C23が登録されており、これらは対応する公開鍵K<sub>P21</sub>乃至K<sub>P23</sub>が含まれている。

#### 【0199】

以上のようなグループを構成する各機器毎に、その証明書が登録された状態において、ユーザからそのグループに属する機器にコンテンツの提供が要求されると、コンテンツサーバ3は、図38のフローチャートに示す処理を実行する。

#### 【0200】

最初に、ステップS281において、コンテンツサーバ3のCPU21は、記憶部28に記憶されている証明書のうち、そのグループに属する証明書を検証する処理を実行する。

#### 【0201】

この検証処理は、図29と図30を参照して説明されたように、各機器の証明書に含まれるリーフIDに基づいて、タグを利用してEKBをたどることで行われる。EKBは、コンテンツサーバ3にも、ライセンスサーバ4から配布されている。この検証処理により、リボーカされている証明書は除外される。

#### 【0202】

ステップS282において、コンテンツサーバ3のCPU21は、ステップS281の検証処理の結果、有効とされた証明書を選択する。そして、ステップS283において、CPU21は、ステップS282の処理で選択された各機器の証明書の各公開鍵でコンテンツ鍵を暗号化する。ステップS284において、CPU21は、対象とされるグループの各機器に、ステップS283の処理で暗号化されたコンテンツ鍵をコンテンツとともに送信する。

#### 【0203】

図37に示されるグループ1のうち、例えば、証明書C14がリボーカされているとすると、ステップS283の処理で、例えば、図39に示されるような暗号化データが生成される。

#### 【0204】

すなわち、図39の例においては、コンテンツ鍵Kcが、証明書C11の公開鍵 $K_{P11}$ 、証明書C12の公開鍵 $K_{P12}$ 、または証明書C13の公開鍵 $K_{P13}$ により、暗号化されている。

#### 【0205】

コンテンツサーバ3の図38に示されるような処理に対応して、コンテンツの提供を受ける各グループの機器（クライアント）は、図40のフローチャートに示す処理を実行する。

#### 【0206】

最初に、ステップS291において、クライアント1のCPU21は、コンテンツサーバ3が図38のステップS284の処理で送信してきたコンテンツを、コンテンツ鍵とともに受信する。コンテンツは、コンテンツ鍵Kcにより、暗号化されており、コンテンツ鍵は上述したように、各機器が保持する公開鍵により暗号化されている（図39）。

## 【0207】

そこで、ステップS292において、CPU21は、ステップS291の処理で受信した自分宛のコンテンツ鍵を、自分自身の秘密鍵で復号し、取得する。そして、取得したコンテンツ鍵を用いてコンテンツの復号処理が行われる。

## 【0208】

例えば、図39の例に示される証明書C11に対応する機器は、公開鍵 $K_{P11}$ に対応する自分自身の秘密鍵を用いて、コンテンツ鍵 $K_c$ の暗号を復号し、コンテンツ鍵 $K_c$ を取得する。そして、コンテンツ鍵 $K_c$ を用いて、コンテンツがさらに復号される。

## 【0209】

同様の処理は、証明書C12, C13に対応する機器においても行われる。リポークされている証明書C14の機器は、自分自身の公開鍵を用いて暗号化されたコンテンツ鍵 $K_c$ がコンテンツに付随して送られてこないので、コンテンツ鍵 $K_c$ を復号することができず、従って、コンテンツ鍵 $K_c$ を用いてコンテンツを復号することができない。

## 【0210】

以上においては、コンテンツキー（すなわちコンテンツ）に対してグルーピングを行うようにしたが、ライセンスキー（ライセンス）に対してグルーピングを行うことも可能である。

## 【0211】

以上のようにして、特別なグループキー、後述するICV (Integrity Check Value) を用いずにグループ化が可能となる。このグループ化は、小規模のグループに適用するのに向いている。

## 【0212】

本発明においては、ライセンスもチェックアウト、あるいはチェックインしたり、ムーブしたり、コピーしたりすることが可能とされる。但し、これらの処理はSDMIで定められたルールに基づいて行われる。

## 【0213】

次に、図41と図42のフローチャートを参照して、このようなクライアント

によるライセンスのチェックアウト処理について説明する。

#### 【0214】

最初に、図4-1のフローチャートを参照して他のクライアントにライセンスをチェックアウトするクライアントの処理について説明する。最初に、ステップS301において、クライアント1のCPU21は、チェックアウト対象のライセンスのチェックアウト回数N1を読み取る。このチェックアウト回数は、図8に示されるように、使用条件に書き込まれているので、この使用条件から読み取られる。

#### 【0215】

次に、ステップS302において、CPU21は、チェックアウト対象のライセンスの最大チェックアウト回数N2を、やはりライセンスの使用条件から読み取る。

#### 【0216】

そして、ステップS303において、CPU21は、ステップS301の処理で読み取られたチェックアウト回数N1と、ステップS302の処理で読み取られた最大チェックアウト回数N2とを比較し、チェックアウト回数N1が最大チェックアウト回数N2より大きいか否かを判定する。

#### 【0217】

チェックアウト回数N1が、最大チェックアウト回数N2より大きいと判定された場合、ステップS304に進み、CPU21は、相手側の装置（チェックアウト先のクライアント）のリーフキーを相手個々の装置から取得し、そのリーフキーを、いまチェックアウト対象とされているライセンスIDに対応して記憶部28のチェックアウトリストに記憶させる。

#### 【0218】

次に、ステップS305において、CPU21は、ステップS301の処理で読み取られたライセンスのチェックアウト回数N1の値を1だけインクリメントする。ステップS306において、CPU21は、ライセンスのメッセージに基づいて、ICVを演算する。このICVについては、図4-6乃至図5-0を参照して後述する。ICVを用いてライセンスの改竄を防止することが可能となる。

## 【0219】

次に、ステップS307において、CPU21は、チェックアウト対象のライセンスと、ステップS306の処理で演算されたICVを、自分自身の公開鍵を用いて暗号化して、EKBおよび証明書とともに、相手側の装置に出力し、コピーさせる。さらに、ステップS308において、CPU21は、ステップS306の処理で演算されたICVを、相手側装置のリーフキーと、ライセンスIDに対応して記憶部28のチェックリスト中に記憶させる。

## 【0220】

ステップS303において、チェックアウト回数N1が最大チェックアウト回数N2より大きくない（例えば、等しい）と判定された場合、もはや許容される回数だけチェックアウトが行われているので、これ以上チェックアウトを行うことができない。そこで、ステップS309に進み、CPU21は、エラー処理を実行する。すなわち、この場合、チェックアウト処理は実行されないことになる。

## 【0221】

次に、図42のフローチャートを参照して、図41のチェックアウト処理により、ライセンスのチェックアウトを受けるクライアントの処理について説明する。

## 【0222】

最初に、ステップS321において、相手側装置（ライセンスをチェックアウトするクライアント1）に、自分自身のリーフキーを送信する。このリーフキーは、ステップS304において、相手側のクライアントにより、ライセンスIDに対応して記憶される。

## 【0223】

次に、ステップS322において、CPU21は、相手側のクライアント1から暗号化されたライセンスとICVが、EKBおよび証明書とともに送信されてきた場合、これを受信する。すなわち、このライセンス、ICV、EKBおよび証明書は、図41のステップS307の処理で相手側の装置から送信されたものである。

## 【0224】

ステップS323において、CPU21は、ステップS322の処理で受信した

ライセンス、ICV、EKBおよび証明書を、記憶部28に記憶させる。

【0225】

以上のようにして、ライセンスのチェックアウトを受けたクライアント1は、チェックアウトを受けたそのライセンスを使用して、所定のコンテンツを再生する場合、図43のフローチャートに示される処理を実行する。

【0226】

すなわち、最初に、ステップS341において、クライアント1のCPU21は、ユーザより入力部26を介して再生が指定されたコンテンツのICVを演算する。そして、ステップS342において、CPU21は、記憶部28に記憶されている暗号化されているICVを、証明書に含まれている公開鍵に基づいて、復号させる。

【0227】

次に、ステップS343において、CPU21は、ステップS341の処理により、いま演算されたICVと、ステップS342の処理により読み出され、復号されたICVが一致するか否かを判定する。両者が一致する場合には、ライセンスは改竄されていないことになる。そこで、ステップS344にすすみ、CPU21は、対応するコンテンツを再生する処理を実行する。

【0228】

これに対して、ステップS343において、2つのICVが一致しないと判定された場合、ライセンスは改竄されている恐れがある。このため、ステップS345に進み、CPU21は、エラー処理を実行する。すなわち、このとき、そのライセンスを用いてコンテンツを再生することができないことになる。

【0229】

次に、以上のようにして、他のクライアントに一旦チェックアウトしたライセンスのチェックインを受けるクライアントの処理について、図44のフローチャートを参照して説明する。

【0230】

最初に、ステップS361において、CPU21は、相手側の装置（ライセンスを返却（チェックイン）してくるクライアント1）のリーフキーと、チェックイ

ン対象のライセンスのIDを取得する。次に、ステップS362において、CPU21は、ステップS361で取得されたチェックイン対象のライセンスが、自分自身が相手側装置にチェックアウトしたライセンスであるか否かを判定する。この判定は、図41のステップS308の処理で記憶されたICV、リーフキー、およびライセンスIDに基づいて行われる。すなわち、ステップS361で取得されたリーフキー、ライセンスID、およびICVが、チェックアウトリスト中に記憶されているか否かが判定され、記憶されている場合には、自分自身がチェックアウトしたライセンスであると判定される。

#### 【0231】

ライセンスが、自分自身がチェックアウトしたものであるとき、ステップS363において、CPU21は、相手側の装置のライセンス、EKBおよび証明書の削除を要求する。後述するように、この要求に基づいて、相手側の装置は、ライセンス、EKBおよび証明書の削除を実行する（図45のステップS383）。

#### 【0232】

ステップS364において、CPU21は、一旦チェックアウトしたライセンスが再びチェックインされてきたので、そのライセンスのチェックアウト回数N1を1だけデクリメントする。

#### 【0233】

ステップS365において、CPU21は、相手側の装置に他のライセンスをチェックアウトしているか否かを判定し、まだチェックアウトしている他のライセンスが存在しない場合には、ステップS365に進み、CPU21は、相手側の装置のチェックイン対象機器としてのチェックアウトリストにおける記憶を削除する。これに対して、ステップS364において、相手側の装置にチェックアウトしている他のライセンスが存在すると判定された場合には、他のライセンスのチェックインを受ける可能性があるので、ステップS366の処理はスキップされる。

#### 【0234】

ステップS362において、チェックイン対象とされているライセンスが、自分自身が相手側装置にチェックアウトしたライセンスではないと判定された場合

、CPU21は、ステップS367に進み、ヘッダ処理を実行する。すなわち、この場合には、自分自身が管轄するライセンスではないことになるので、チェックイン処理は実行されない。

#### 【0235】

ユーザが、ライセンスを不正にコピーしたような場合、記憶されているICVの値と、ステップS361の処理で取得されたライセンスに基づいて演算されたICVの値が異なるものとなるで、チェックインできることになる。

#### 【0236】

図45は、図44のフローチャートに示されるライセンスのチェックイン処理を実行するクライアントに対して、自分自身が有しているライセンスをチェックインさせるクライアントの処理を表している。

#### 【0237】

ステップS381において、クライアント1のCPU21は、相手側の装置（図44のフローチャートに示す処理を実行するクライアント1）にリーフキーとチェックイン対象のライセンスのIDを送信する。上述したように、相手側の装置は、ステップS361において、このリーフキーとライセンスIDを取得し、ステップS362において、それに基づいて、チェックイン対象のライセンスの認証処理を実行する。

#### 【0238】

ステップS382において、クライアント1のCPU21は、相手側の装置からライセンスの削除を要求されたか否かを判定する。すなわち、ライセンスが正当なチェックイン対象のライセンスである場合、上述したように、相手側の装置は、ステップS363の処理でライセンス、EKBおよび証明書の削除を要求してくれる。そこで、この要求を受信した場合、ステップS383に進み、CPU21は、ライセンス、EKBおよび証明書を削除する。すなわち、これにより、このクライアント1は、以後そのライセンスを使用できない状態となり、図44のステップS364の処理により、チェックアウト回数N1が、1だけデクリメントされるので、チェックインが完了したことになる。

#### 【0239】

ステップS382において、相手側の装置からライセンスの削除が要求されていないと判定された場合、ステップS384に進み、エラー処理が実行される。すなわち、この場合には、ICVの値が異なっている等の理由により、チェックインができないことになる。

## 【0240】

以上においては、チェックインとチェックアウトについて説明したが、同様に、ライセンスをコピーあるいはムーブするようにすることも可能である。

## 【0241】

次に、ライセンス（コンテンツも同様）の改竄を防止するためにライセンスのインテグリティ・チェック値（ICV）を生成して、ライセンスに対応付けて、ICVの計算により、ライセンス改竄の有無を判定する処理構成について説明する。

## 【0242】

ライセンスのインテグリティ・チェック値（ICV）は、例えばライセンスに対するハッシュ関数を用いて計算され、 $ICV = \text{hash}(K_{icv}, L_1, L_2, \dots)$ によって計算される。 $K_{icv}$ はICV生成キーである。 $L_1, L_2$ はライセンスの情報であり、ライセンスの重要な情報のメッセージ認証符号（MAC : Message authentication Code）が使用される。

## 【0243】

DES暗号処理構成を用いたMAC値生成例を図46に示す。図46の構成に示すように対象となるメッセージを8バイト単位に分割し、（以下、分割されたメッセージをM1、M2、・・・、MNとする）、まず、初期値（IV）とM1を、演算部24-1Aにより排他的論理和する（その結果をI1とする）。次に、I1をDES暗号化部24-1Bに入れ、鍵（以下、K1とする）を用いて暗号化する（出力をE1とする）。続けて、E1およびM2を演算部24-2Aにより排他的論理和し、その出力I2をDES暗号化部24-2Bへ入れ、鍵K1を用いて暗号化する（出力をE2）。以下、これを繰り返し、全てのメッセージに対して暗号化処理を施す。DES暗号化部24-NBから最後に出てきたENがメッセージ認証符号（MAC（Message Authentication Code））となる。

## 【0244】

このようなライセンスのMAC値とICV生成キーにハッシュ関数を適用してライセンスのインテグリティ・チェック値(ICV)が生成される。例えばライセンス生成時に生成したICVと、新たにライセンスに基づいて生成したICVとを比較して同一のICVが得られればライセンスに改竄のないことが保証され、ICVが異なれば、改竄があったと判定される。

## 【0245】

次に、ライセンスのインテグリティ・チェック値(ICV)生成キーであるK<sub>i c v</sub>を上述の有効化キーブロックによって送付する構成について説明する。すなわちEKBによる暗号化メッセージデータをライセンスのインテグリティ・チェック値(ICV)生成キーとした例である。

## 【0246】

図4.7および図4.8に複数のデバイスに共通のライセンスを送付した場合、それらのライセンスの改竄の有無を検証するためのインテグリティ・チェック値生成キーK<sub>i c v</sub>を有効化キーブロック(EKB)によって配信する構成例を示す。図4.7はデバイス0, 1, 2, 3に対して復号可能なチェック値生成キーK<sub>i c v</sub>を配信する例を示し、図4.8はデバイス0, 1, 2, 3中のデバイス3をリボーカ(排除)してデバイス0, 1, 2に対してのみ復号可能なチェック値生成キーK<sub>i c v</sub>を配信する例を示す。

## 【0247】

図4.7の例では、更新ノードキーK(t)00によって、チェック値生成キーK<sub>i c v</sub>を暗号化したデータE<sub>n c</sub>(K(t)00, K<sub>i c v</sub>)とともに、デバイス0, 1, 2, 3においてそれぞれの有するノードキー、リーフキーを用いて更新されたノードキーK(t)00を復号可能な有効化キーブロック(EKB)を生成して配信する。それぞれのデバイスは、図4.7の右側に示すように、まず、EKBを処理(復号)することにより、更新されたノードキーK(t)00を取得し、次に、取得したノードキーK(t)00を用いて、暗号化されたチェック値生成キーE<sub>n c</sub>(K(t)00, K<sub>i c v</sub>)を復号して、チェック値生成キーK<sub>i c v</sub>を得ることが可能となる。

## 【0248】

その他のデバイス4, 5, 6, 7 . . . は同一の有効化キーブロック（EKB）を受信しても自身の保有するノードキー、リーフキーでは、EKBを処理して更新されたノードキー $K(t)_{00}$ を取得することができないので、安全に正当なデバイスに対してのみチェック値生成キーを送付することができる。

## 【0249】

一方、図48の例は、図12の点線枠で囲んだグループにおいてデバイス3が、例えば鍵の漏洩によりリボーカ（排除）されているとして、他のグループのメンバ、すなわち、デバイス0, 1, 2, に対してのみ復号可能な有効化キーブロック（EKB）を生成して配信した例である。図48に示す有効化キーブロック（EKB）と、チェック値生成キー（Kicv）をノードキー（K(t)00）で暗号化したデータEnc（K(t)00, Kicv）を配信する。

## 【0250】

図48の右側には、復号手順を示してある。デバイス0, 1, 2は、まず、受領した有効化キーブロックから自身の保有するリーフキーまたはノードキーを用いた復号処理により、更新ノードキー（K(t)00）を取得する。次に、K(t)00による復号によりチェック値生成キーKicvを取得する。

## 【0251】

図12に示す他のグループのデバイス4, 5, 6 . . . は、この同様のデータ（EKB）を受信したとしても、自身の保有するリーフキー、ノードキーを用いて更新ノードキー（K(t)00）を取得することができない。同様にリボーカされたデバイス3においても、自身の保有するリーフキー、ノードキーでは、更新ノードキー（K(t)00）を取得することができず、正当な権利を有するデバイスのみがチェック値生成キーを復号して利用することが可能となる。

## 【0252】

このように、EKBを利用したチェック値生成キーの配送を用いれば、データ量を少なくして、かつ安全に正当権利者のみが復号可能としたチェック値生成キーを配信することが可能となる。

## 【0253】

このようなライセンスのインテグリティ・チェック値（ICV）を用いることにより、EKBと暗号化ライセンスの不正コピーを排除することができる。例えば図49Aに示すように、ライセンスL1とライセンスL2とをそれぞれのライセンスキーオブジェクトを取得可能な有効化キーブロック（EKB）とともに格納したメディア1があり、これをそのままメディア2にコピーした場合を想定する。EKBと暗号化ライセンスのコピーは可能であり、これをEKBを復号可能なデバイスでは利用できることになる。

#### 【0254】

図49Bに示す例では、各メディアに正当に格納されたライセンスに対応付けてインテグリティ・チェック値（ICV（L1, L2））を格納する構成とする。なお、（ICV（L1, L2））は、ライセンスL1とライセンスL2にハッシュ関数を用いて計算されるライセンスのインテグリティ・チェック値である $ICV = \text{hash}(\text{Kicv}, L1, L2)$ を示している。図49Bの構成において、メディア1には正当にライセンス1とライセンス2が格納され、ライセンスL1とライセンスL2に基づいて生成されたインテグリティ・チェック値（ICV（L1, L2））が格納される。また、メディア2には正当にライセンス1が格納され、ライセンスL1に基づいて生成されたインテグリティ・チェック値（ICV（L1））が格納される。

#### 【0255】

この構成において、メディア1に格納された{EKB, ライセンス2}をメディア2にコピーしたとすると、メディア2で、ライセンスチェック値を新たに生成すると、ICV（L1, L2）が生成されることになり、メディア2に格納されているKicv（L1）と異なり、ライセンスの改竄あるいは不正なコピーによる新たなライセンスの格納が実行されたことが明らかになる。メディアを再生するデバイスにおいて、再生ステップの前ステップにICVチェックを実行して、生成ICVと格納ICVの一致を判別し、一致しない場合は、再生を実行しない構成とすることにより、不正コピーのライセンスの再生を防止することが可能となる。

#### 【0256】

また、さらに、安全性を高めるため、ライセンスのインテグリティ・チェック値（ICV）を書き換えカウンタを含めたデータに基づいて生成する構成としてもよい。すなわち  $ICV = \text{hash}(\text{Kicv}, \text{counter} + 1, L1, L2, \dots)$  によって計算する構成とする。ここで、カウンタ（counter + 1）は、ICVの書き換えごとに1つインクリメントされる値として設定する。なお、カウンタ値はセキュアなメモリに格納する構成とする必要である。

#### 【0257】

さらに、ライセンスのインテグリティ・チェック値（ICV）をライセンスと同一メディアに格納することができない構成においては、ライセンスのインテグリティ・チェック値（ICV）をライセンスとは別のメディア上に格納する構成としてもよい。

#### 【0258】

例えば、読み込み専用メディアや通常のMO等のコピー防止策のとられていないメディアにライセンスを格納する場合、同一メディアにインテグリティ・チェック値（ICV）を格納すると ICV の書き換えが不正なユーザによりなされる可能性があり、ICV の安全性が保てないおそれがある。この様な場合、ホストマシン上の安全なメディアに ICV を格納して、ライセンスのコピーコントロール（例えば check-in/check-out、move）に ICV を使用する構成とすることにより、ICV の安全な管理およびライセンスの改竄チェックが可能となる。

#### 【0259】

この構成例を図50に示す。図50では読み込み専用メディアや通常のMO等のコピー防止策のとられていないメディア2201にライセンス1乃至ライセンス3が格納され、これらのライセンスに関するインテグリティ・チェック値（ICV）を、ユーザが自由にアクセスすることの許可されないホストマシン上の安全なメディア2202に格納し、ユーザによる不正なインテグリティ・チェック値（ICV）の書き換えを防止した例である。このような構成として、例えばメディア2201を装着したデバイスが、メディア2201の再生を実行する際にホストマシンであるPC、サーバにおいて ICV のチェックを実行して再生の可

否を判定する構成とすれば、不正なコピーライセンスあるいは改竄ライセンスの再生を防止できる。

【0260】

本発明が適用されるクライアントは、いわゆるパーソナルコンピュータ以外に、PDA (Personal Digital Assistants) 、携帯電話機、ゲーム端末機などとすることができる。

【0261】

一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、ネットワークや記録媒体からインストールされる。

【0262】

この記録媒体は、図2に示されるように、装置本体とは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク41 (フロッピディスクを含む) 、光ディスク42 (CD-ROM(Compact Disk Read Only Memory), DVD(Digital Versatile Disk)を含む) 、光磁気ディスク43 (MD (Mini-Disk) を含む) 、もしくは半導体メモリ44などよりなるパッケージメディアにより構成されるだけでなく、装置本体に予め組み込まれた状態でユーザに提供される、プログラムが記録されているROM22や、記憶部28に含まれるハードディスクなどで構成される。

【0263】

なお、本明細書において、記録媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0264】

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【0265】

【発明の効果】

以上の如く、本発明の情報処理装置および方法、記録媒体、並びにプログラムによれば、第1のノードより下位の階層の第3のノードに対応するデバイスノードキーが割り当てられているデバイスに、第2のノードより下位の階層の第4のノードに対応するデバイスノードキーを供給するようにしたので、1つのデバイスで、異なるカテゴリのコンテンツを利用することが可能となり、操作性が改善される。

【図面の簡単な説明】

【図1】

本発明を適用したコンテンツ提供システムの構成を示すブロック図である。

【図2】

図1のクライアントの構成を示すブロック図である。

【図3】

図1のクライアントのコンテンツのダウンロード処理を説明するフローチャートである。

【図4】

図1のコンテンツサーバのコンテンツ提供処理を説明するフローチャートである。

【図5】

図4のステップS26におけるフォーマットの例を示す図である。

【図6】

図1のクライアントのコンテンツ再生処理を説明するフローチャートである。

【図7】

図6のステップS43のライセンス取得処理の詳細を説明するフローチャートである。

【図8】

ライセンスの構成を示す図である。

【図9】

図1のライセンスサーバのライセンス提供の処理を説明するフローチャートである。

【図10】

図6のステップS4.5におけるライセンス更新処理の詳細を説明するフローチャートである。

【図11】

図1のライセンスサーバのライセンス更新処理を説明するフローチャートである。

【図12】

キーの構成を説明する図である。

【図13】

カテゴリノードを説明する図である。

【図14】

ノードとデバイスの対応の具体例を示す図である。

【図15】

有効化キーブロックの構成を説明する図である。

【図16】

有効化キーブロックの利用を説明する図である。

【図17】

有効化キーブロックのフォーマットの例を示す図である。

【図18】

有効化キーブロックのタグの構成を説明する図である。

【図19】

DNKを用いたコンテンツの復号処理を説明する図である。

【図20】

有効化キーブロックの例を示す図である。

【図21】

複数のコンテンツの1つのデバイスに対する割り当てを説明する図である。

【図22】

ライセンスのカテゴリを説明する図である。

【図23】

クライアントのリッピング処理を説明するフローチャートである。

【図24】

ウォーターマークの構成を説明する図である。

【図25】

コンテンツのフォーマットの例を示す図である。

【図26】

公開鍵証明書の例を示す図である。

【図27】

コンテンツの配布を説明する図である。

【図28】

クライアントのコンテンツのチェックアウト処理を説明するフローチャートである。

【図29】

タグによる有効化キーブロックをたどる例を説明する図である。

【図30】

有効化キーブロックの構成例を示す図である。

【図31】

マークの構成を説明する図である。

【図32】

クライアントのライセンス買い取り処理を説明するフローチャートである。

【図33】

ライセンスサーバのライセンス買い取り処理を説明するフローチャートである

【図34】

マークの構成例を示す図である。

【図35】

クライアントの証明書の登録処理を説明するフローチャートである。

【図36】

コンテンツサーバの証明書登録処理を説明するフローチャートである。

【図37】

グループの証明書の例を示す図である。

【図38】

グルーピングが行われている場合におけるコンテンツサーバの処理を説明するフローチャートである。

【図39】

コンテンツキーの暗号化の例を示す図である。

【図40】

グループに属するクライアントの処理を説明するフローチャートである。

【図41】

他のクライアントにライセンスをチェックアウトするクライアントの処理を説明するフローチャートである。

【図42】

他のクライアントからライセンスのチェックアウトを受けるクライアントの処理を説明するフローチャートである。

【図43】

ライセンスのチェックアウトを受けたクライアントの再生処理を説明するフローチャートである。

【図44】

他のクライアントからライセンスのチェックインを受けるクライアントの処理を説明するフローチャートである。

【図45】

他のクライアントにライセンスをチェックインするクライアントの処理を説明するフローチャートである。

【図46】

MACの生成を説明する図である。

【図47】

ICV生成キーの復号処理を説明するフローチャートである。

【図48】

ICV生成キーの他の復号処理を説明する図である。

【図49】

ICVによるライセンスのコピーの管理を説明する図である。

【図50】

ライセンスの管理を説明する図である。

【符号の説明】

1-1, 1-2 クライアント, 2 インターネット, 3 コンテンツサーバ,  
4 ライセンスサーバ, 5 課金サーバ, 20 タイマ, 21  
CPU, 24 暗号化復号部, 25 コーデック部, 26 入力部, 27  
出力部, 28 記憶部, 29 通信部

【書類名】図面

【図1】

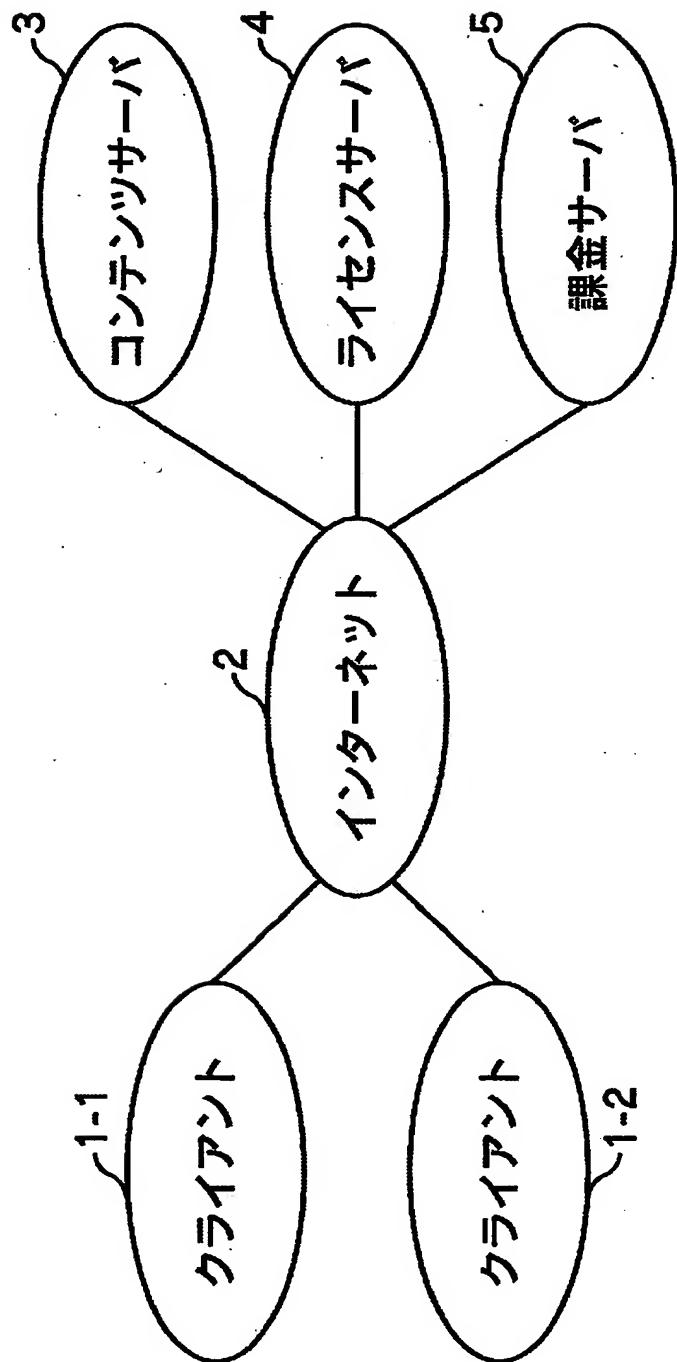


図1

【図2】

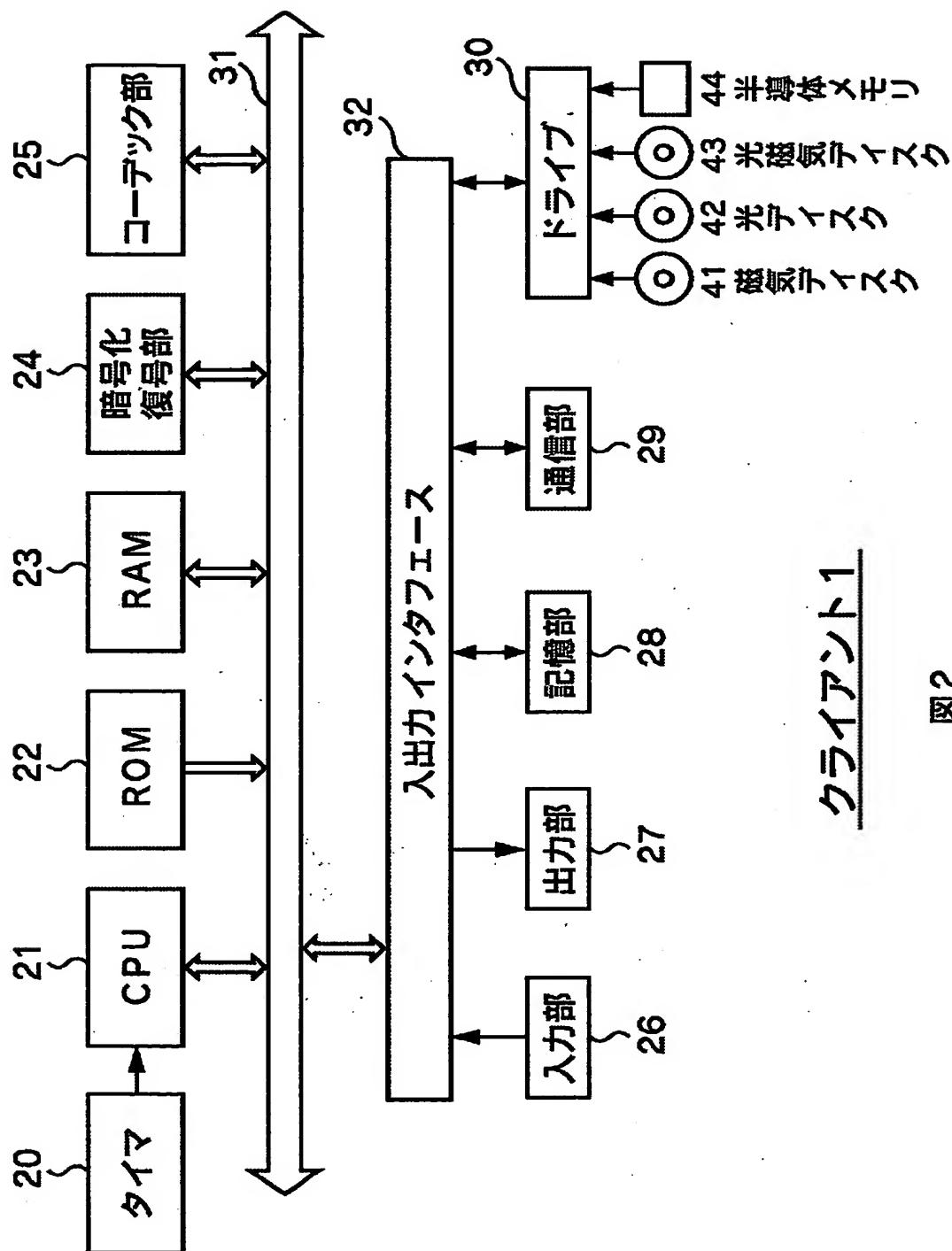
クライアント1

図2

【図3】

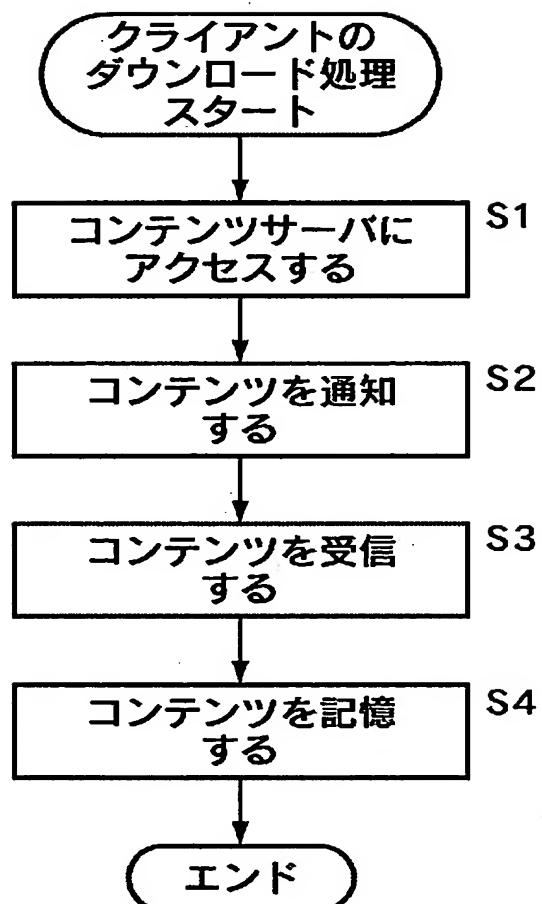


図3

【図4】

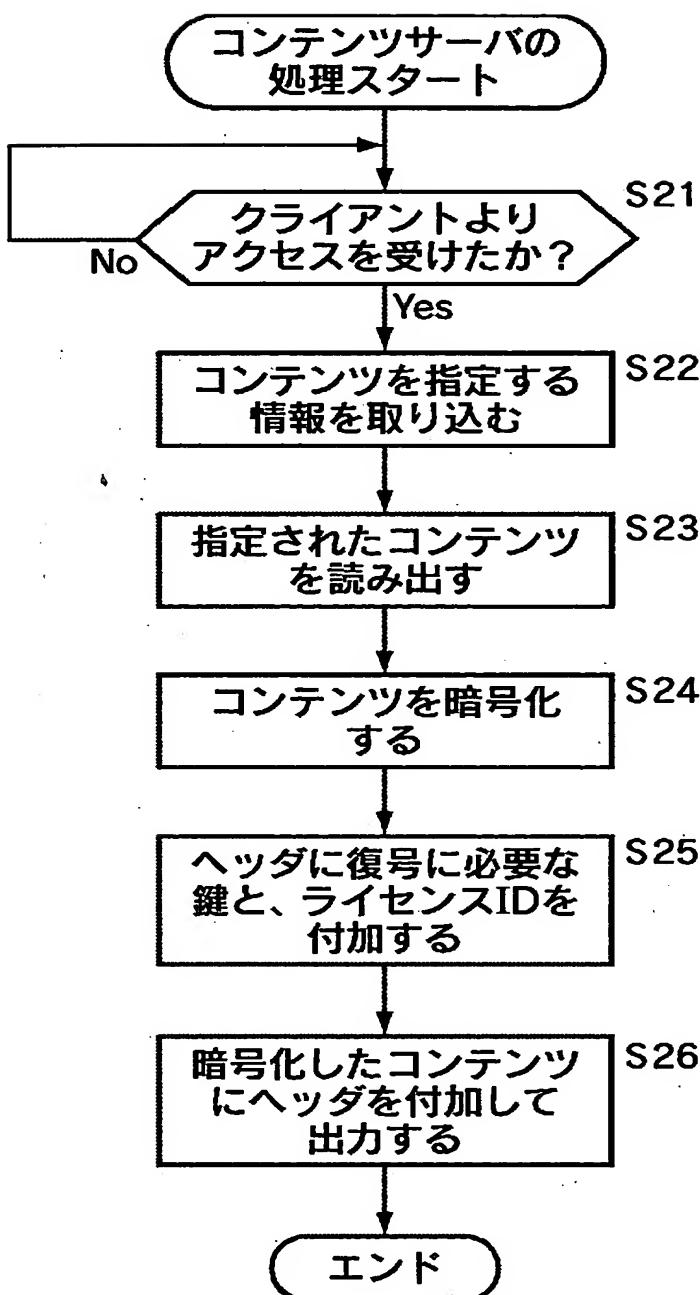


図4

【図5】

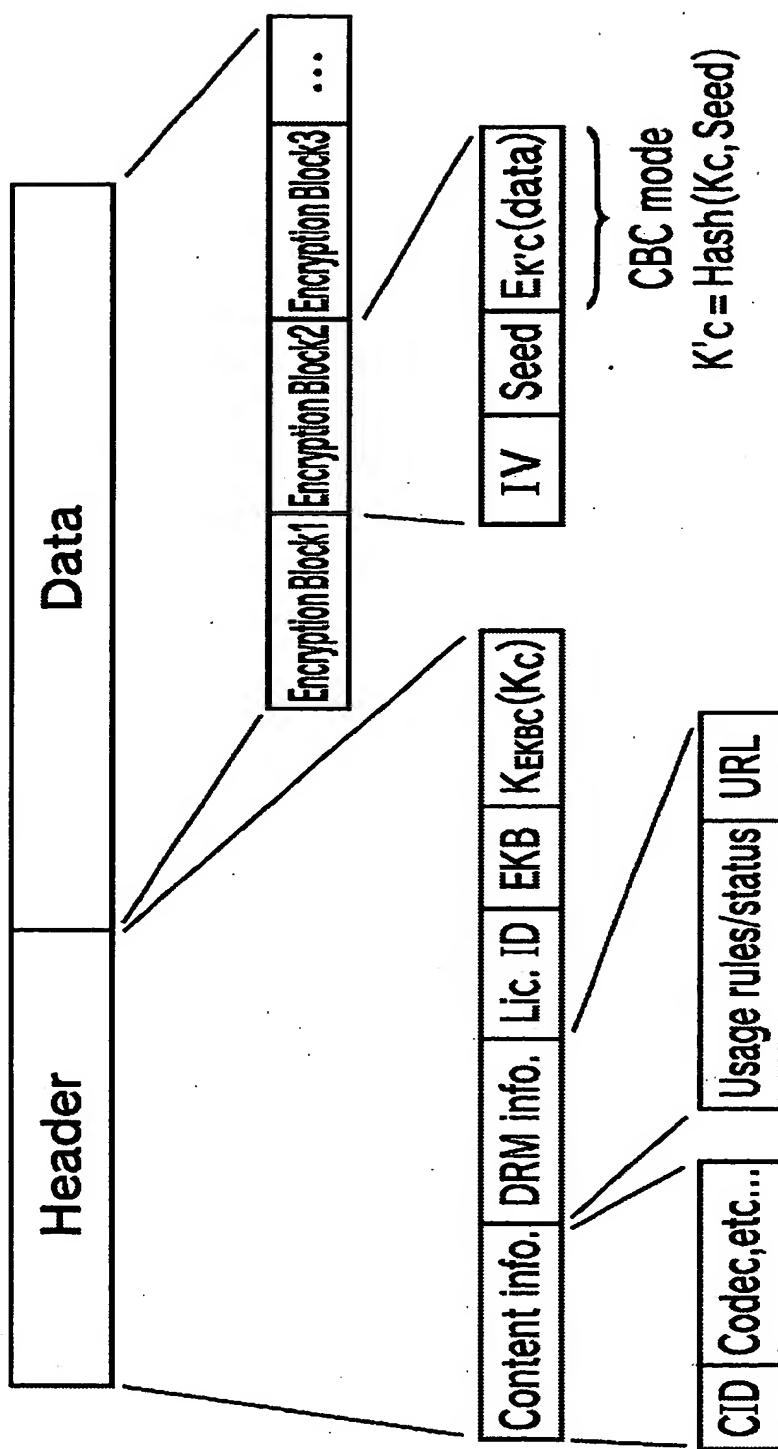


図5

【図6】

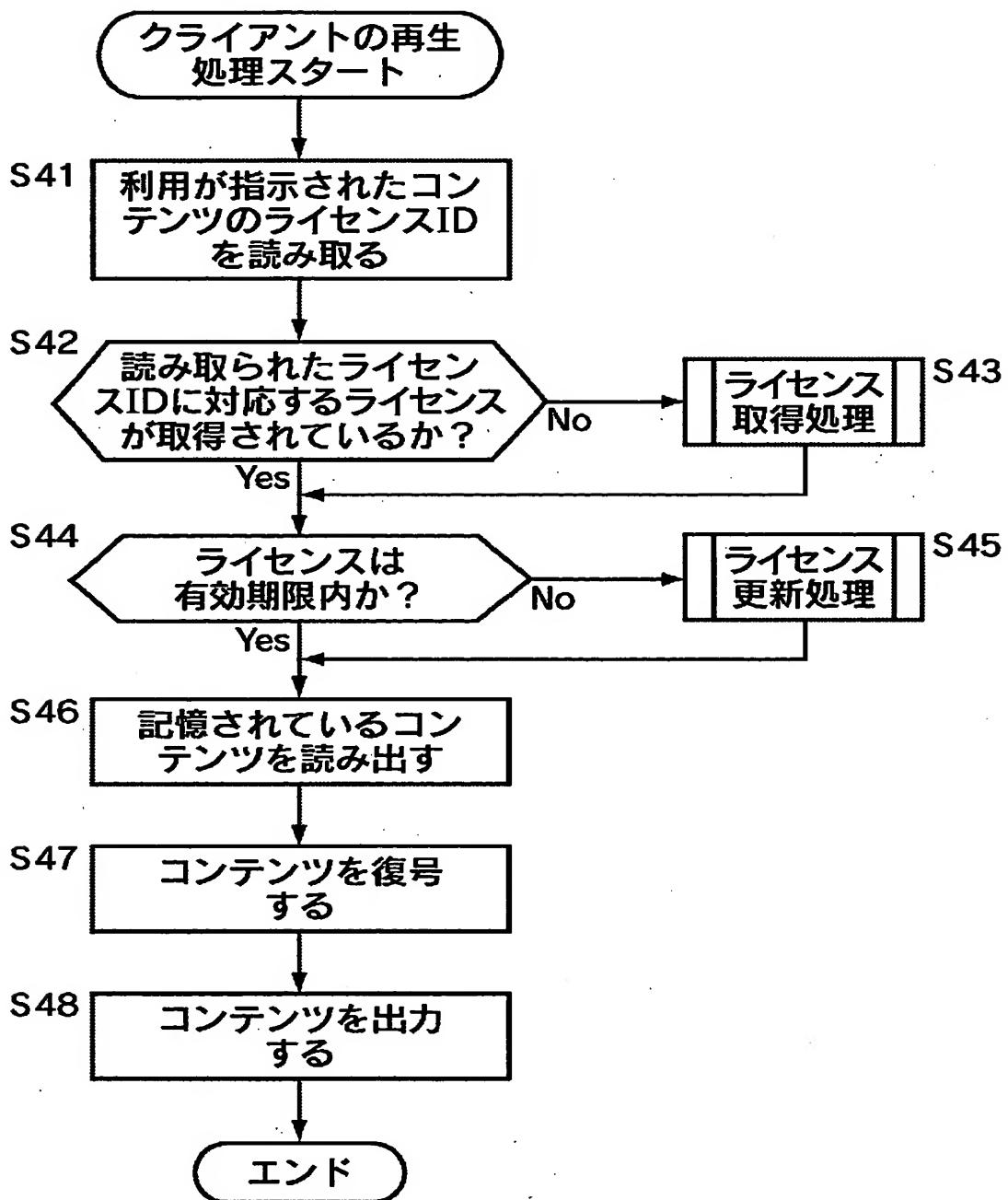


図6

【図7】

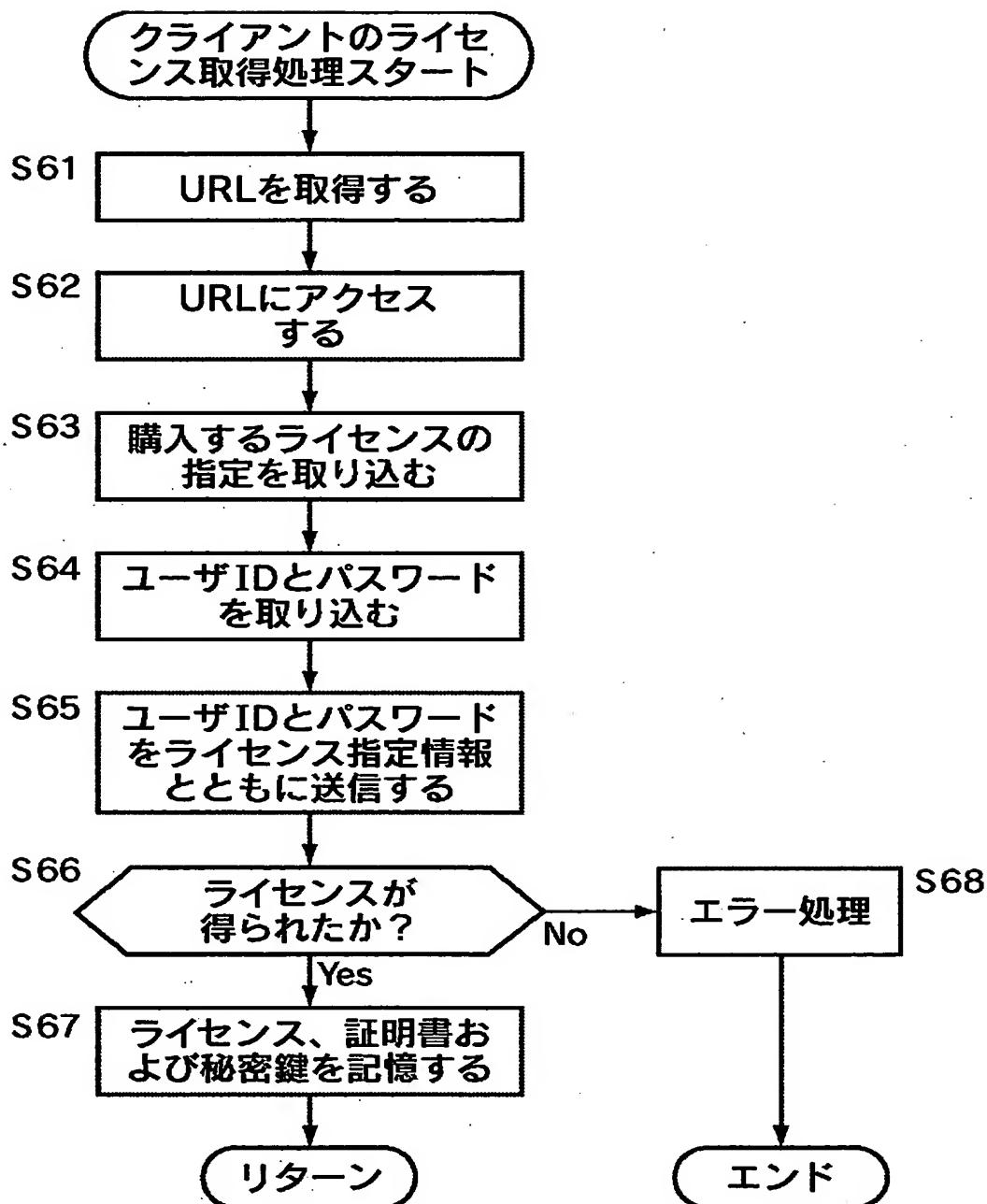


図7

【図8】

使 用 条 件	使用期限	CD-Rに記録可能な権利
	ダウンロード期限	PDにコピー可能な回数
	許されるコピー回数	所有権に移行できる権利
	チェックアウト回数	使用ログを取る義務
	チェックアウト可能回数	
リーフID		
DNK		

## ライセンス

図8

【図9】

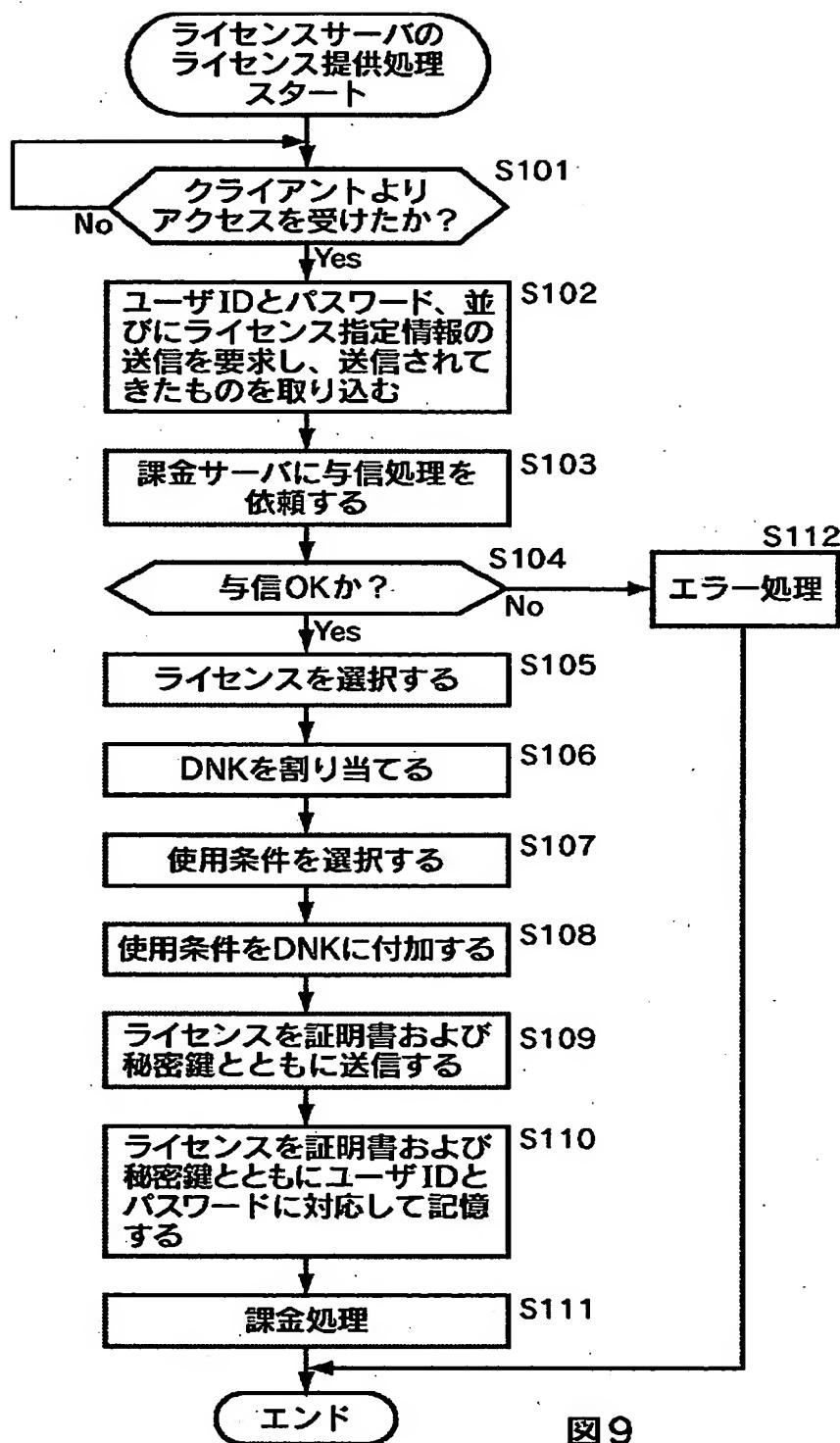


図9

【図10】

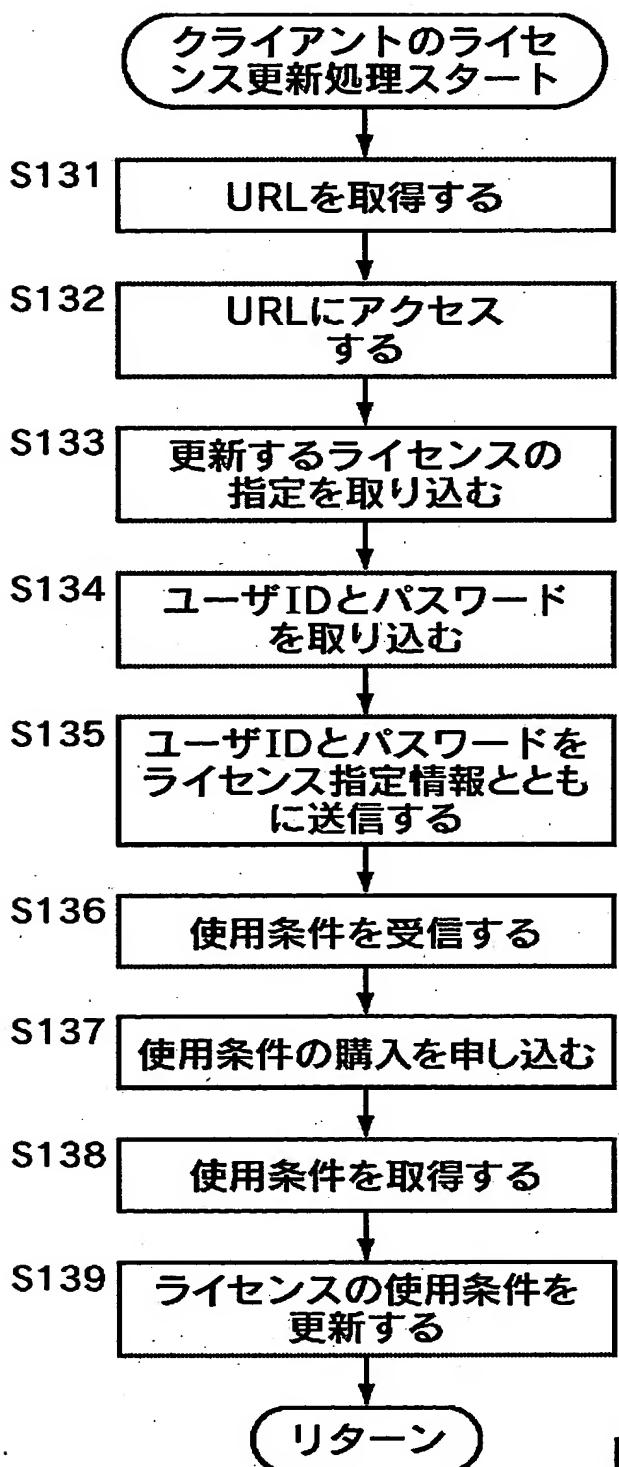


図10

【図11】

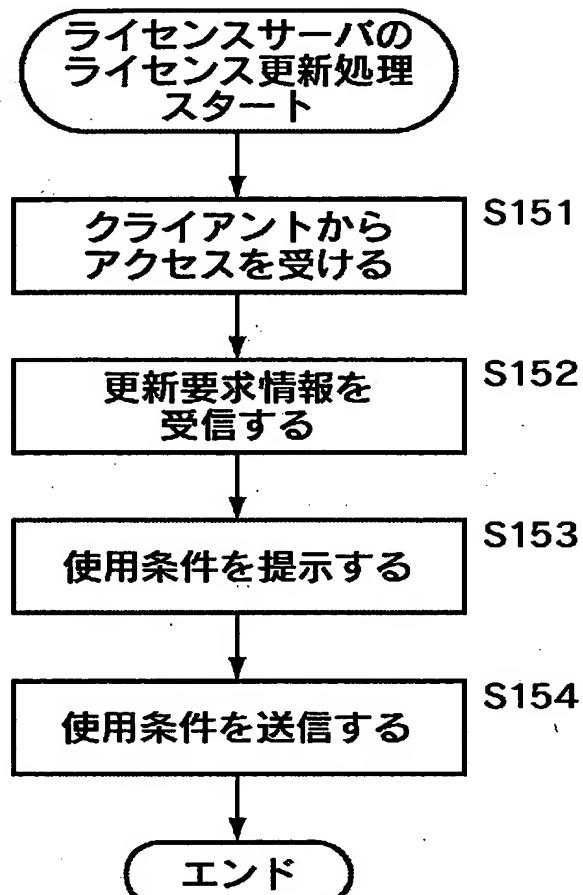


図11

【図12】

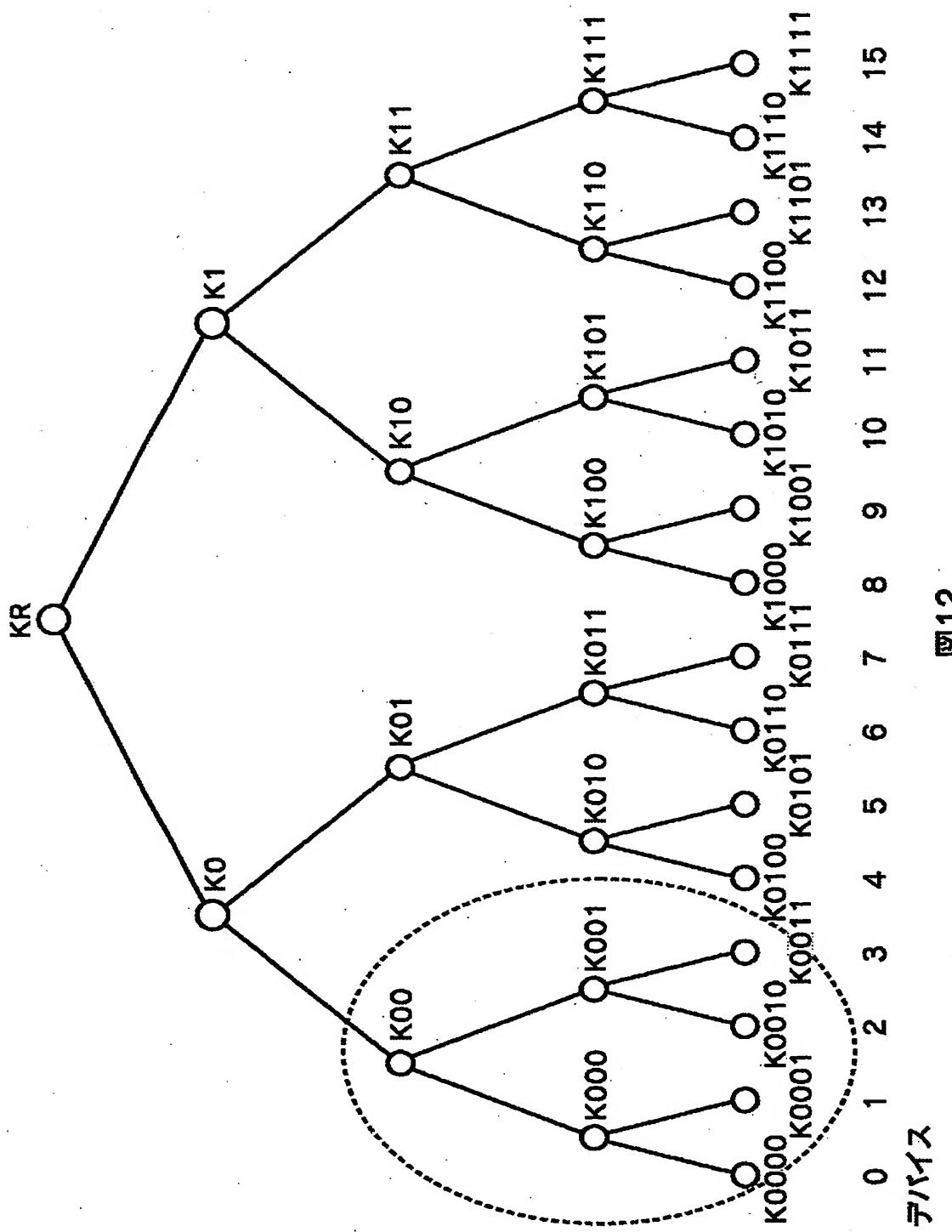


図12

【図13】

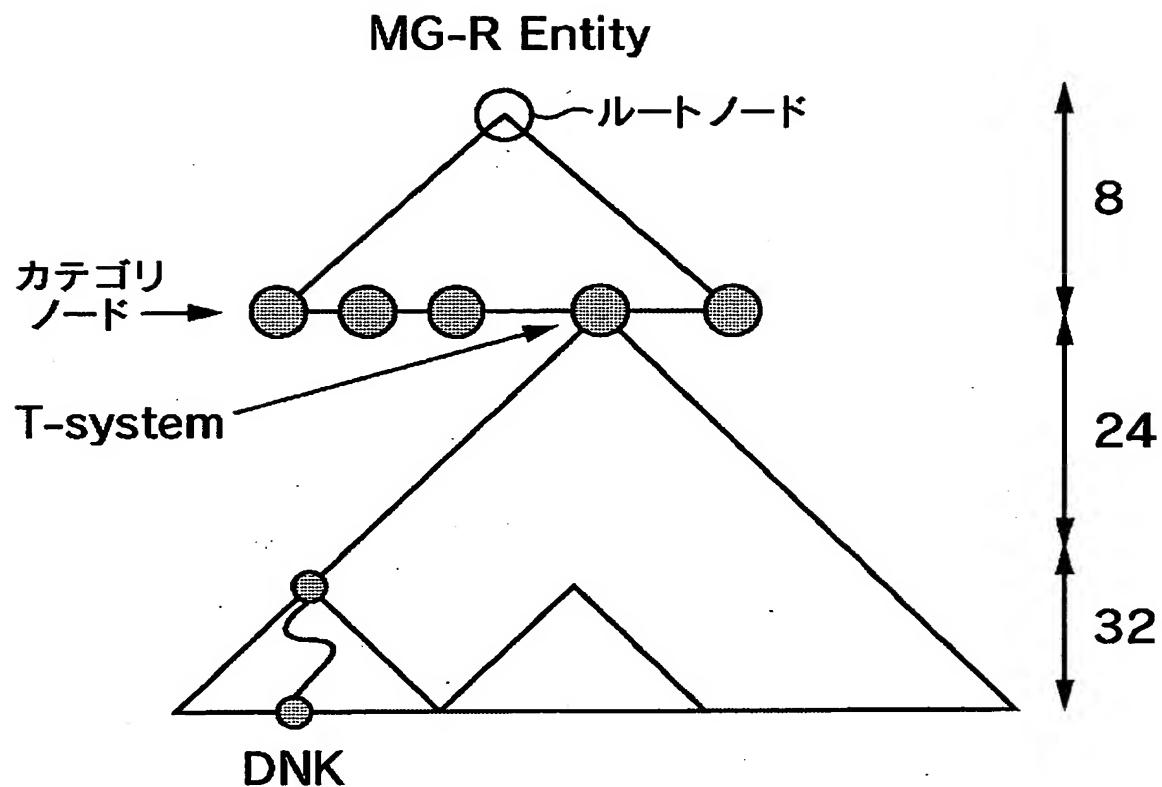


図13

【図14】

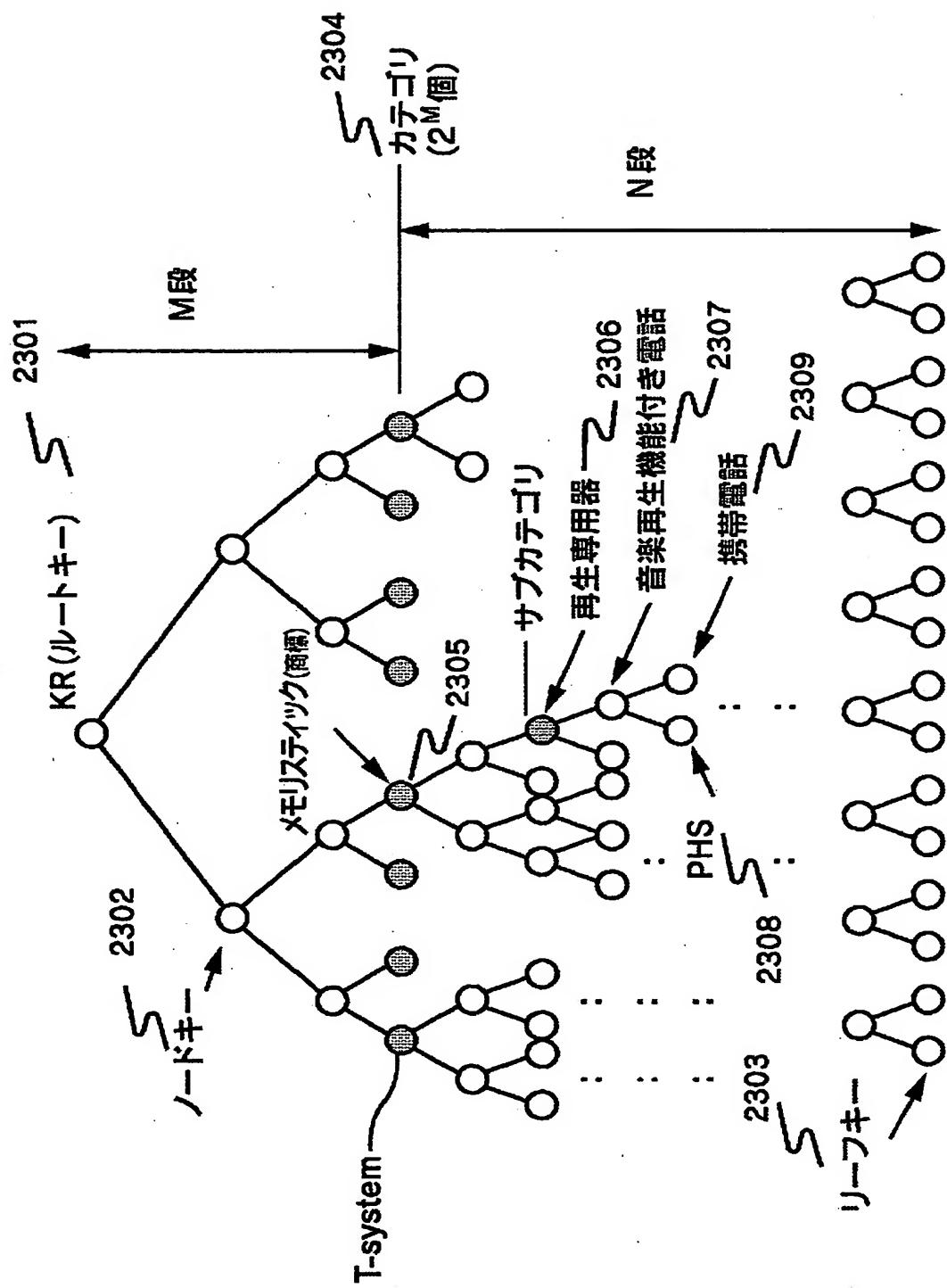


図14

【図15】

- A 有効化キーブロック(EKB:Enabling Key Block)  
デバイス0,1,2にバージョン:t のノードキーを送付

バージョン(Version) :t	
インデックス	暗号化キー
0	Enc(K(t)0, K(t)R)
00	Enc(K(t)00, K(t)0)
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

- B 有効化キーブロック(EKB:Enabling Key Block)  
デバイス0,1,2にバージョン:t のノードキーを送付

バージョン(Version) :t	
インデックス	暗号化キー
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

図15

【図16】

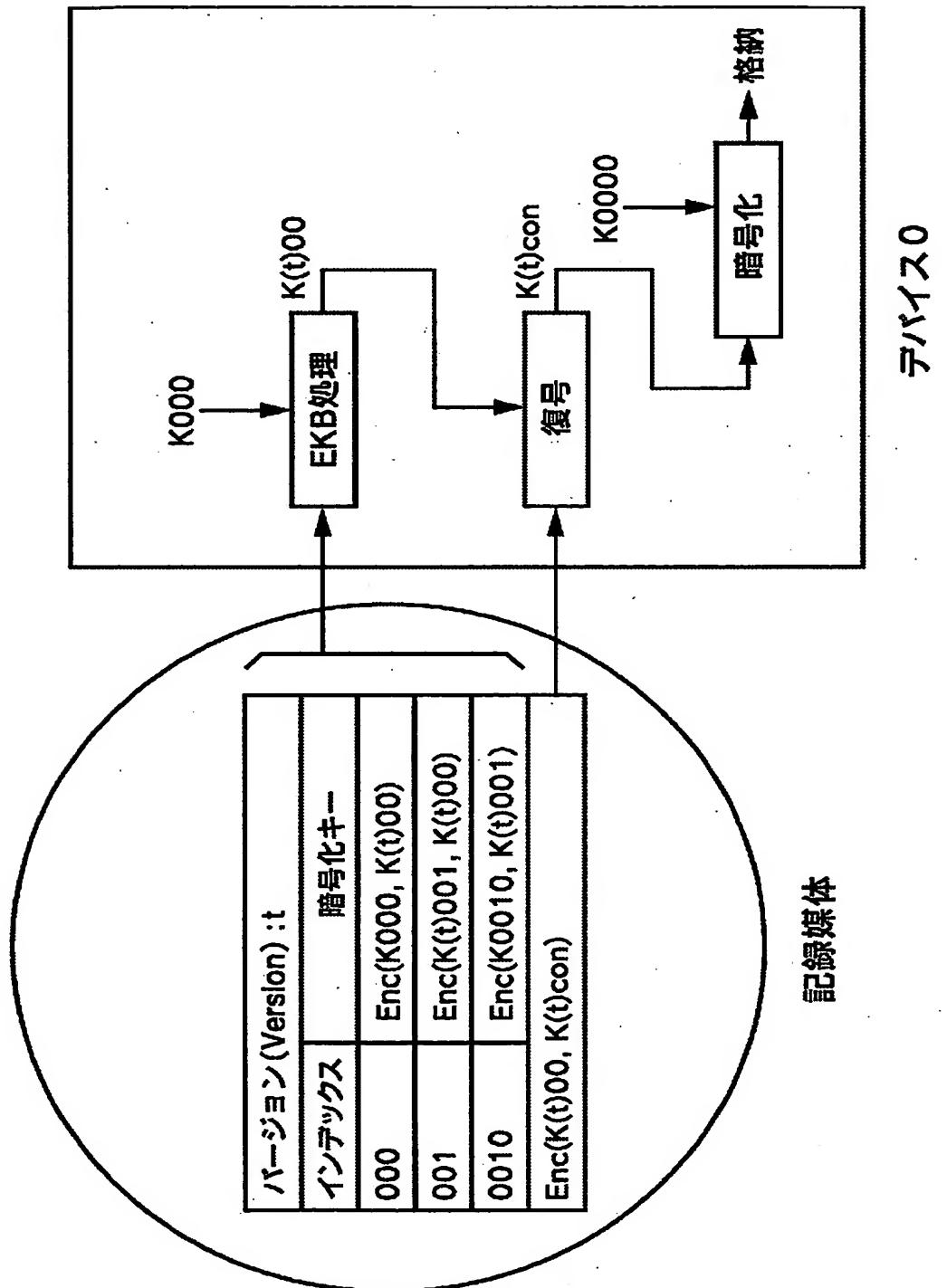
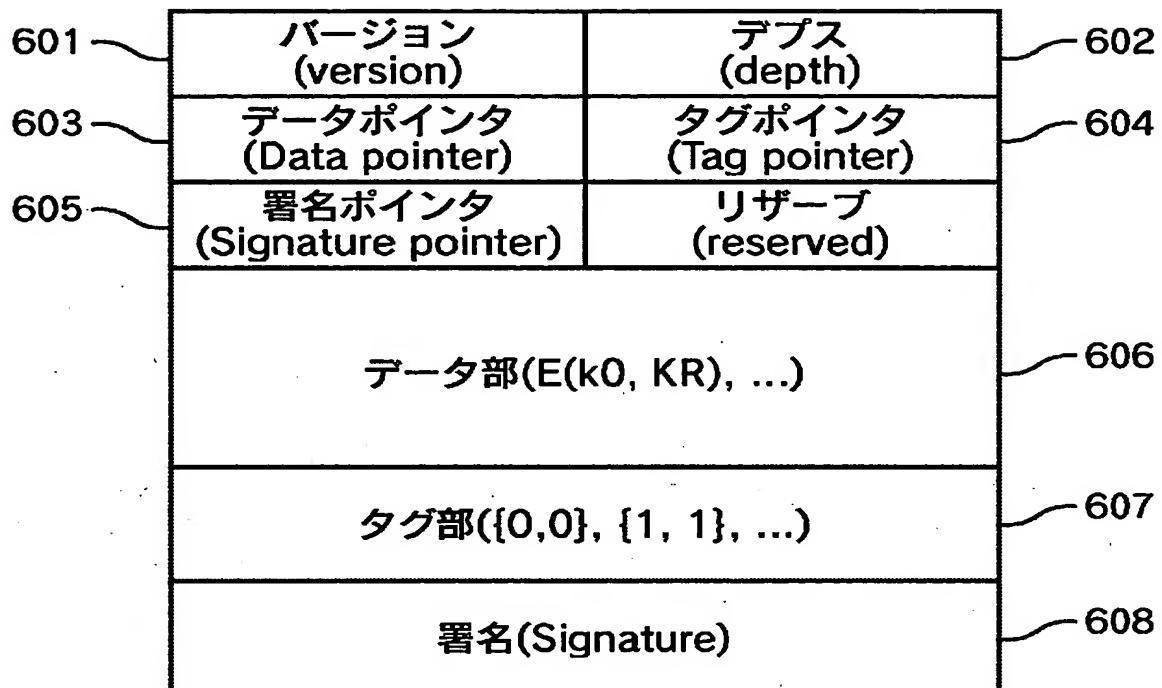


図16

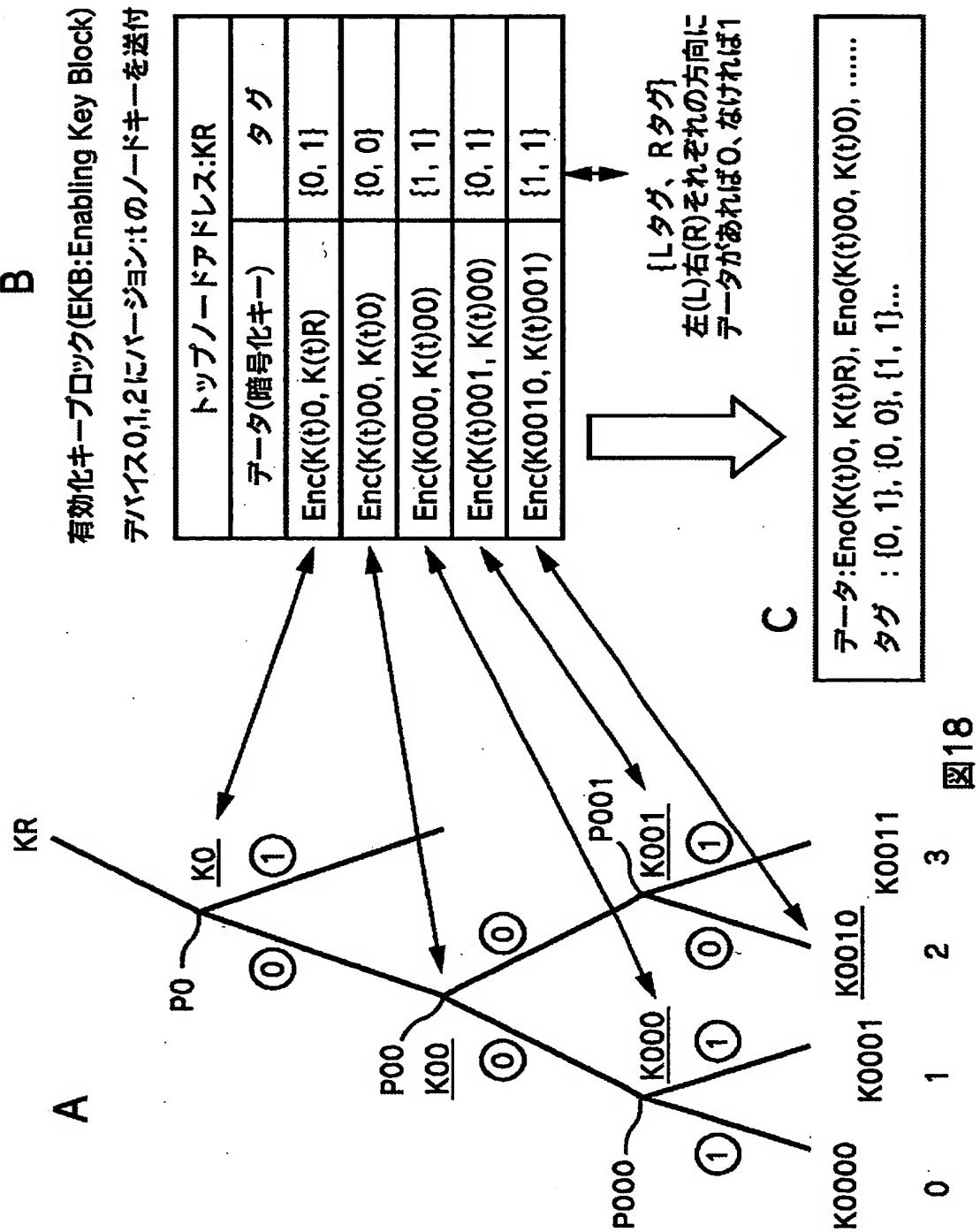
【図17】



EKB

図17

【図18】



【図19】

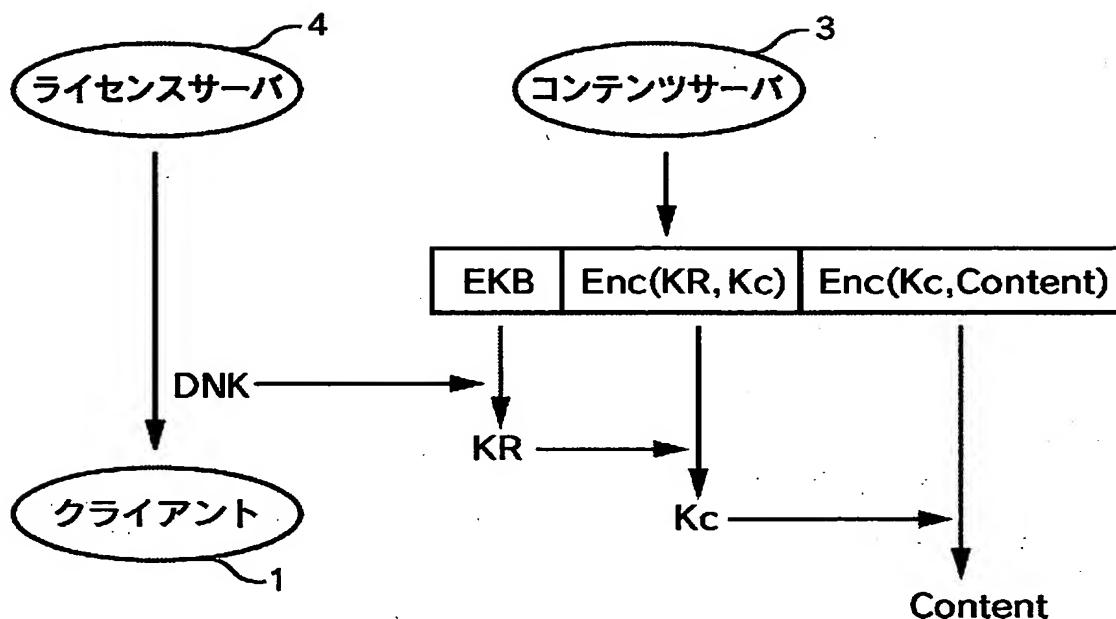


図19

【図20】

**EKB****Enc(DNK, KR)**

図20

【図21】

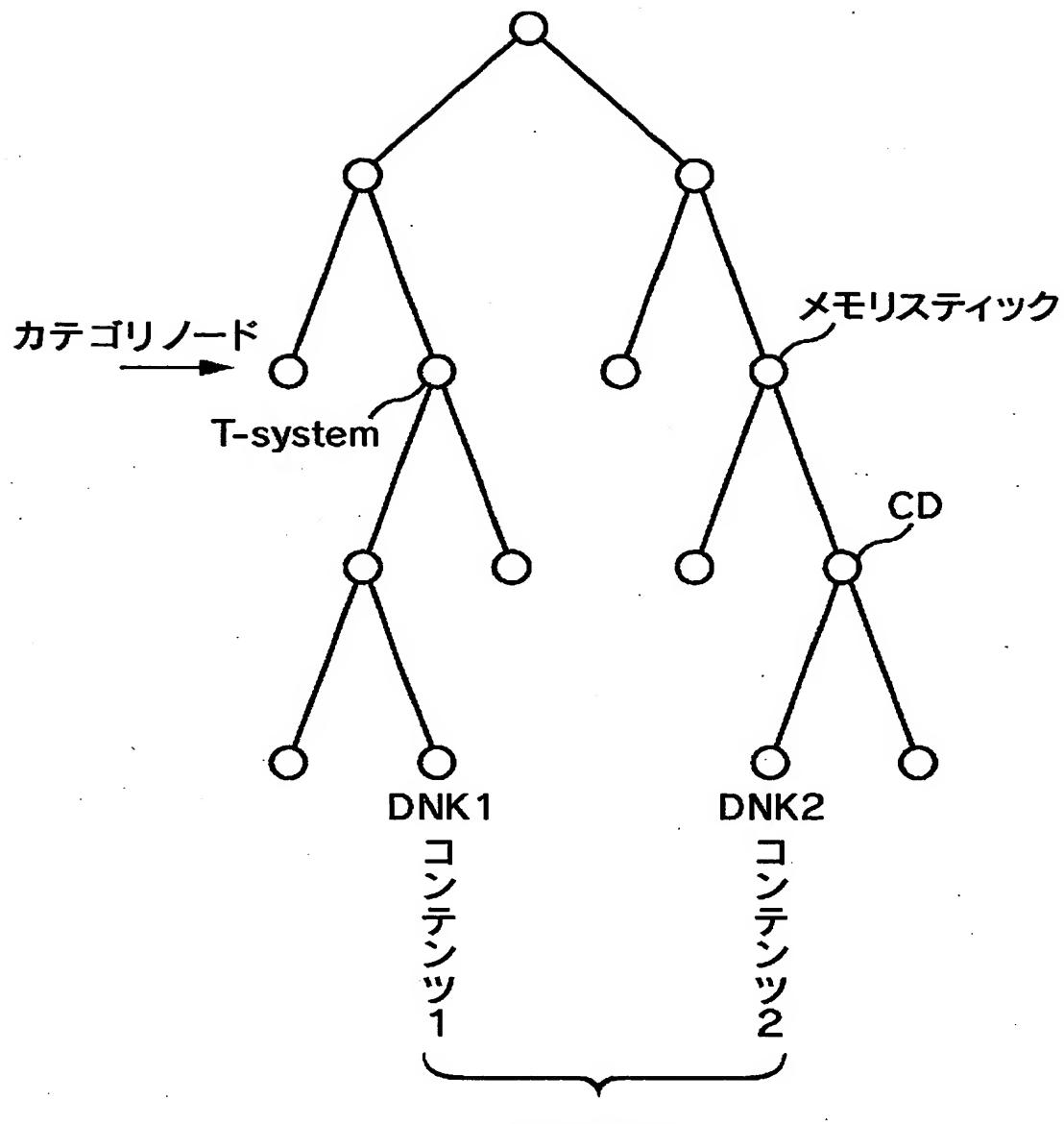


図21

【図2.2】

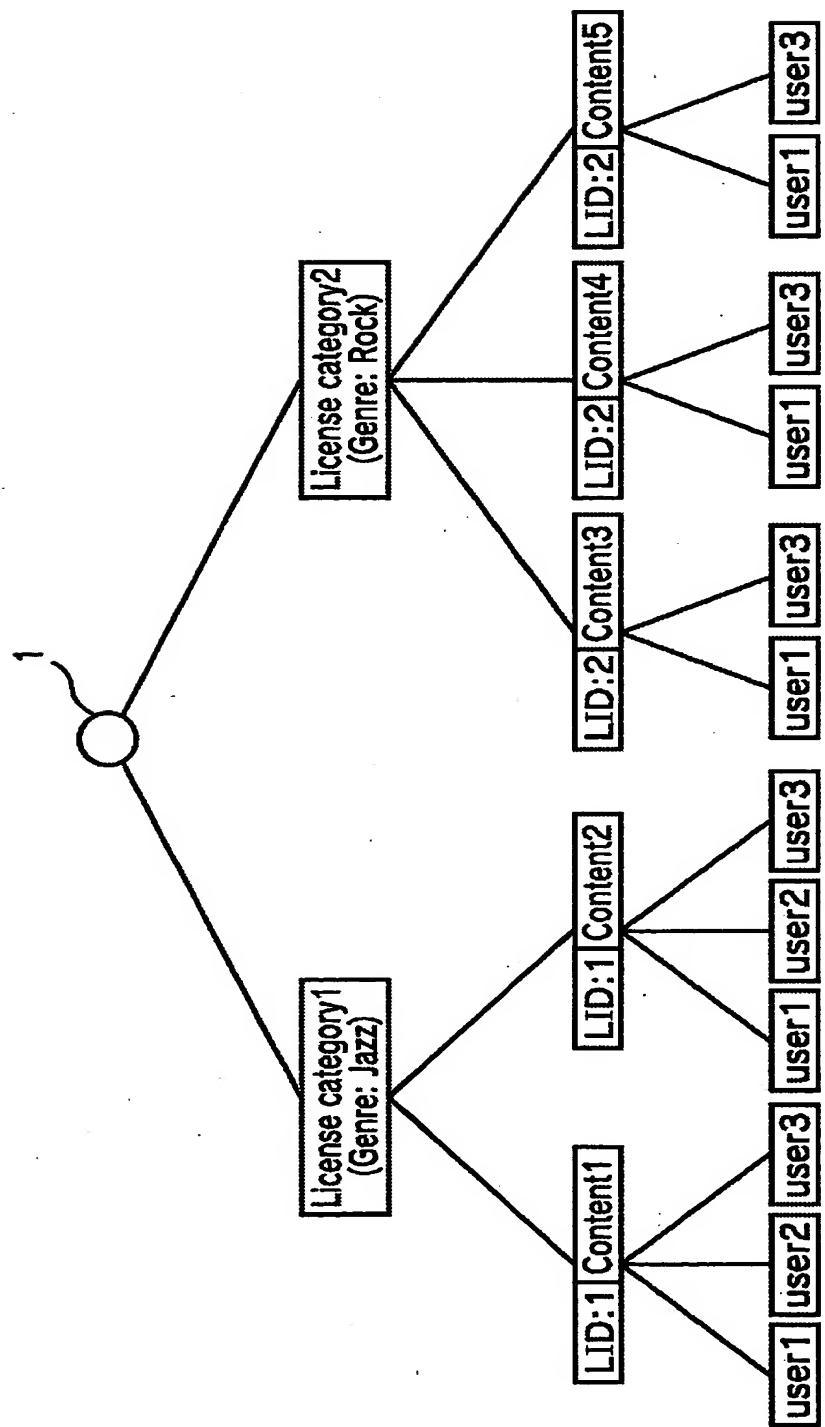


図22

【図23】

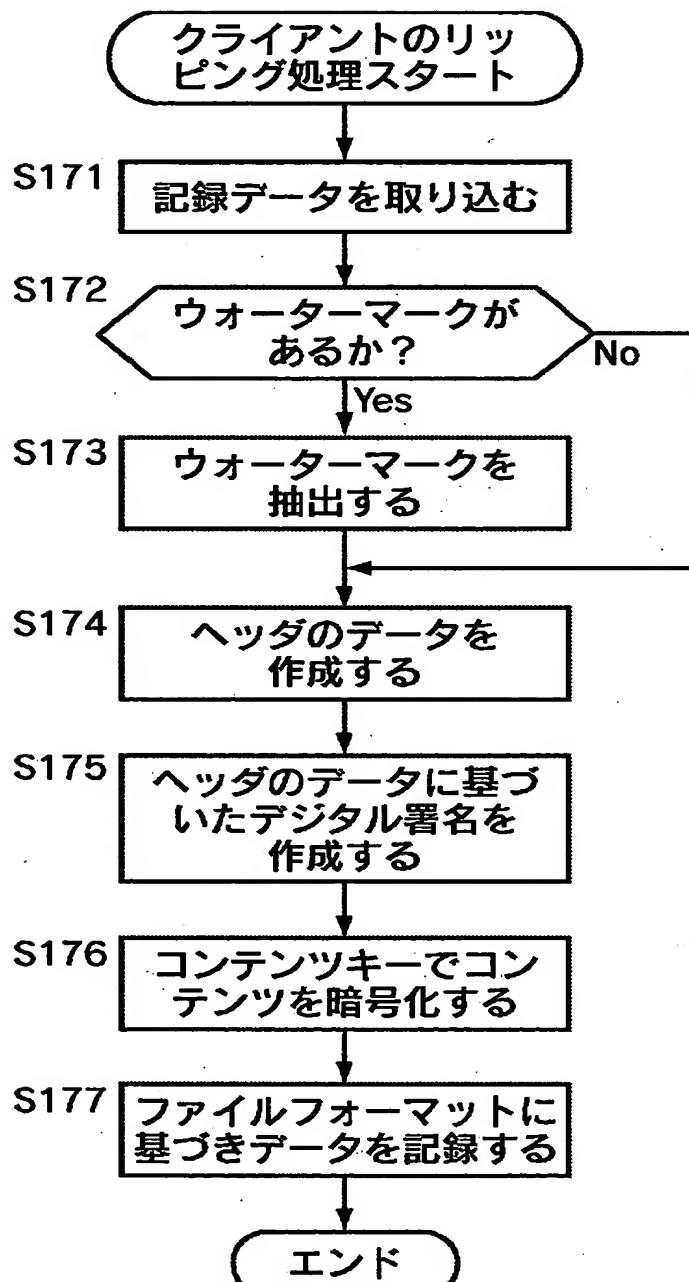


図23

【図24】

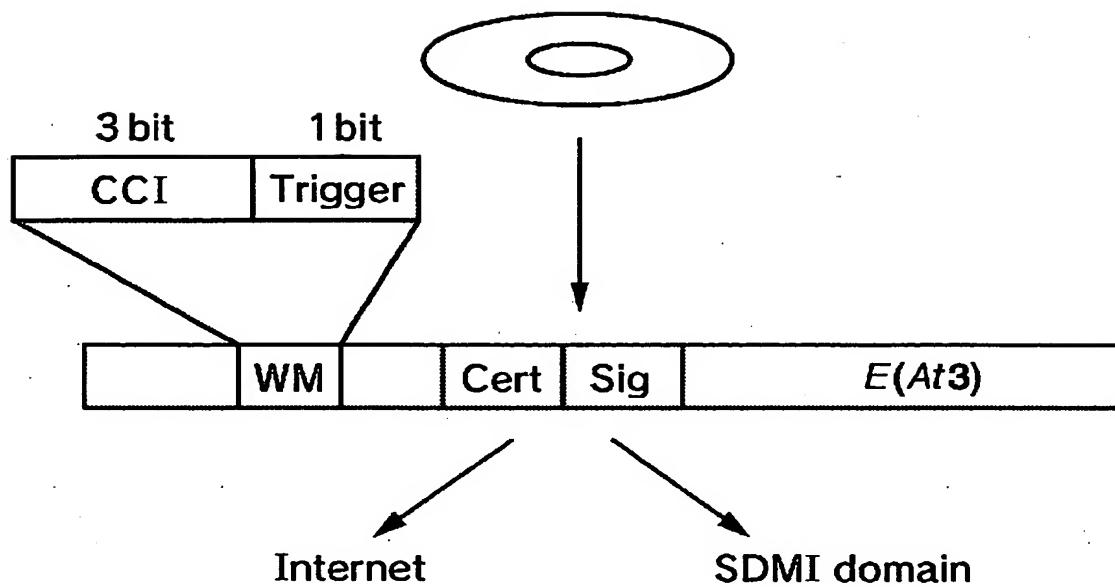


図24

【図25】

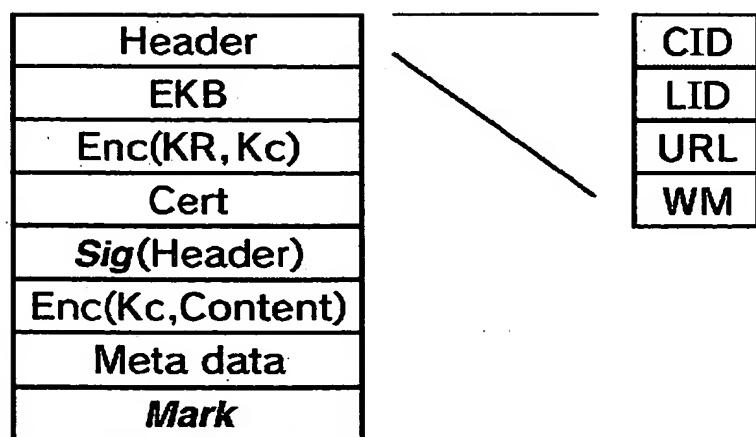
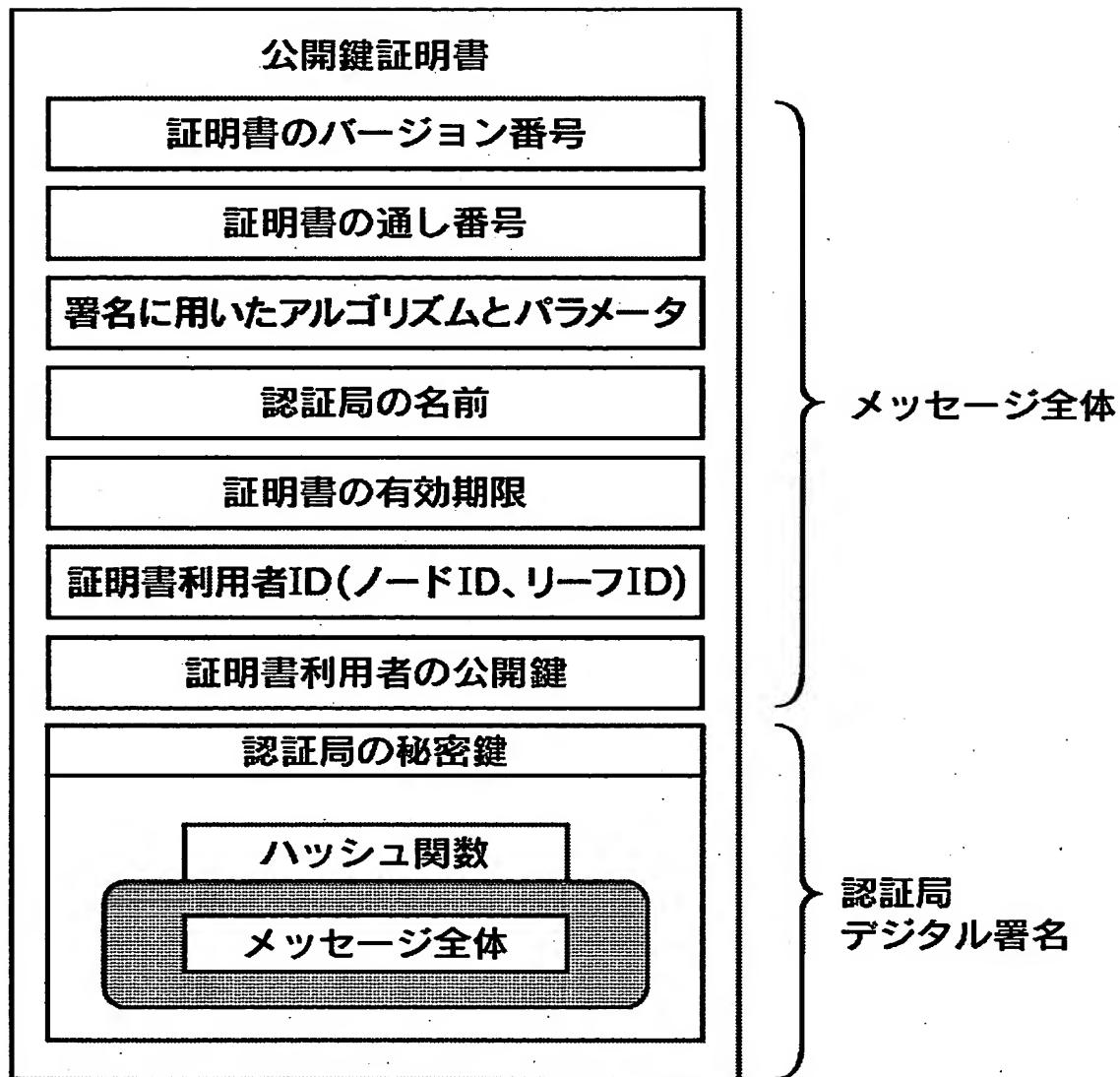


図25

【図26】



## 公開鍵証明書

図26

【図27】

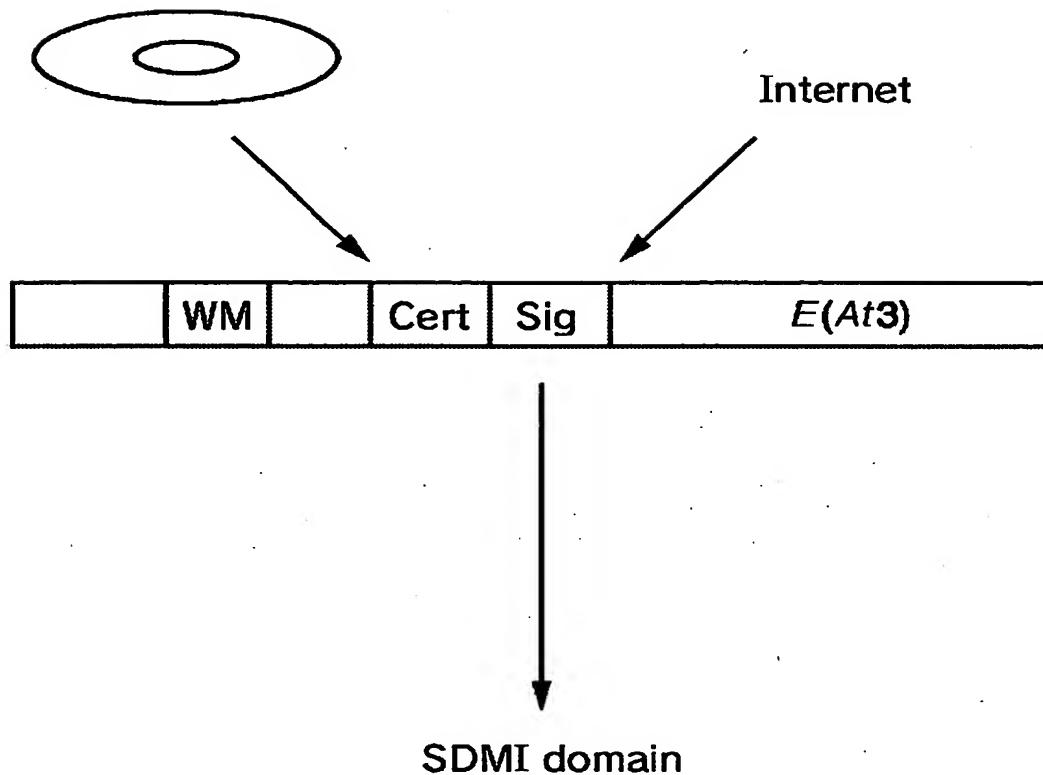


図27

【図28】

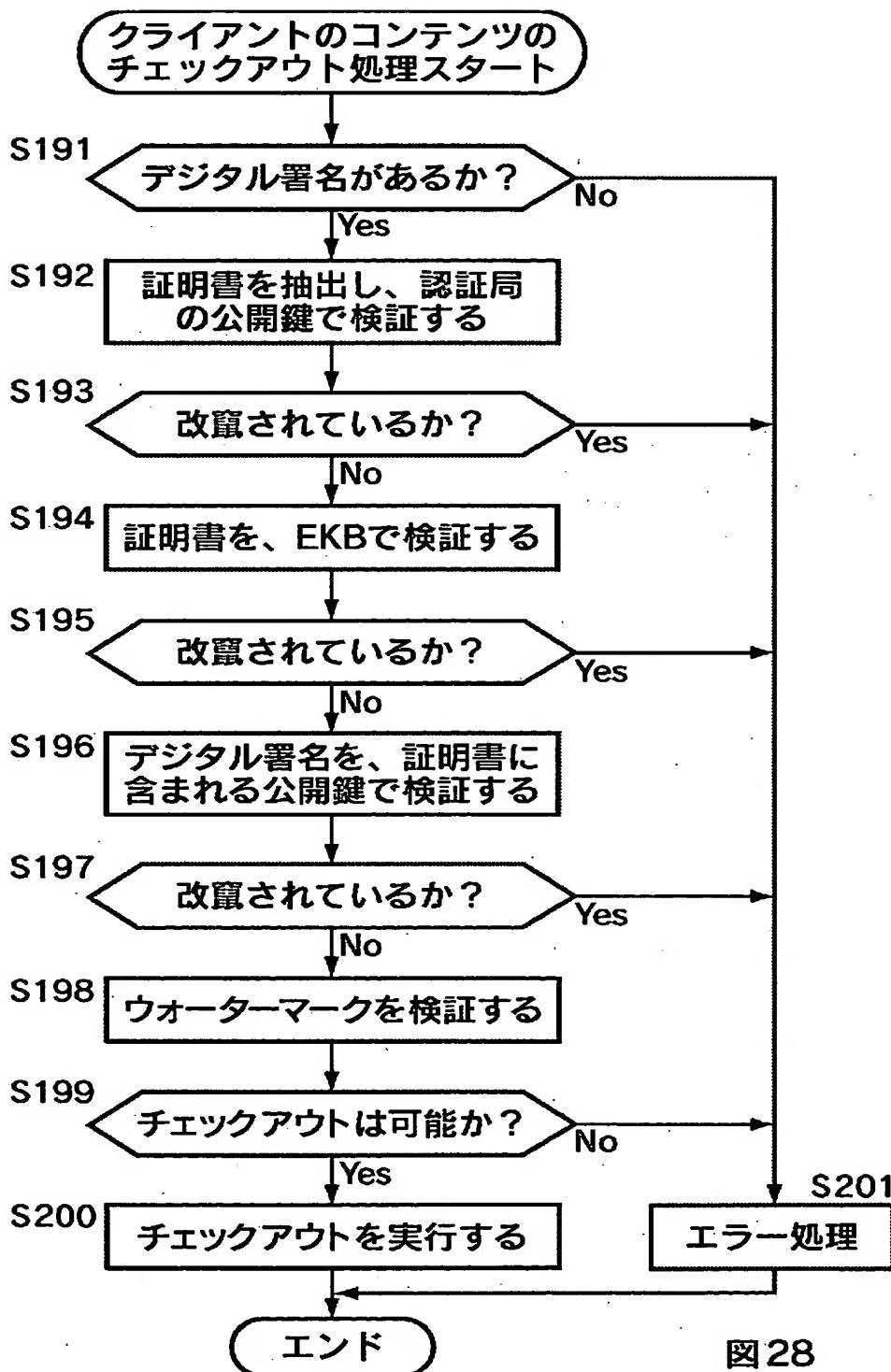


図28

【図29】

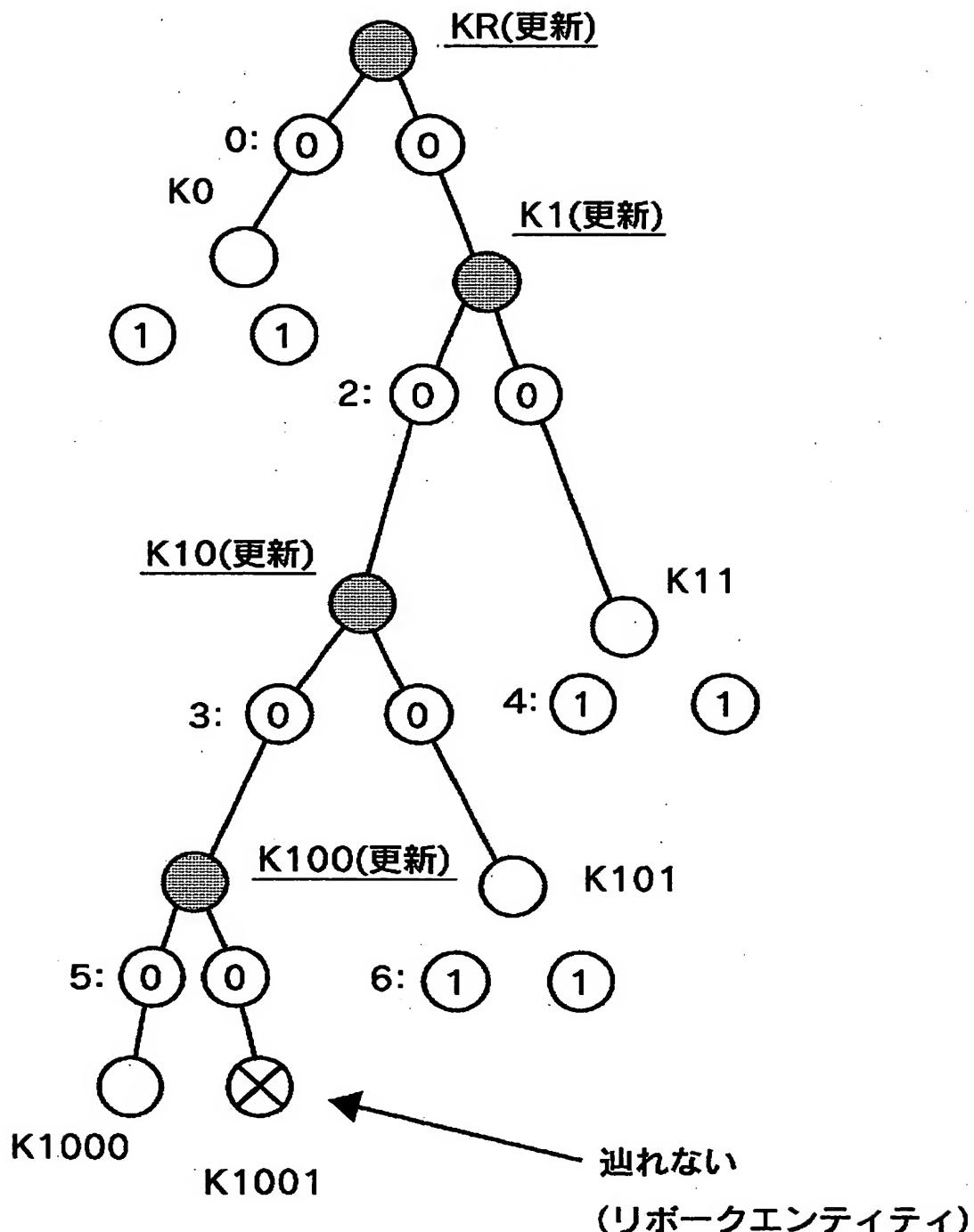


図29

【図30】

有効化キーブロック(EKB:Enabling Key Block)の  
データ部およびタグ

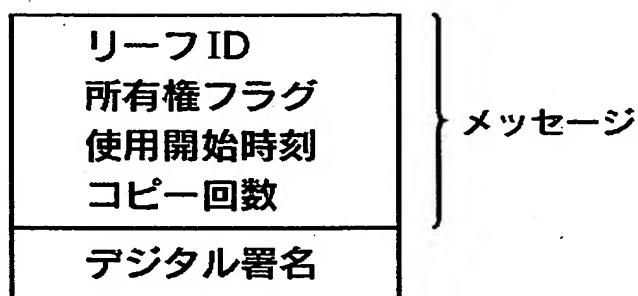
データ (暗号化キー)	Enc(K0, K(t)R), Enc(K(t)1, K(t)R) Enc(K(t)10, K(t)1), Enc(K11, K(t)1) Enc(K(t)100, K(t)10), Enc(K101, K(t)10) Enc(K1000, K(t)100)
タグ	0: {0, 0}, 1:{1, 1}, 2:{0, 0}, 3:(0, 0) 4: {1, 1}, 5:{0, 1}, 6:{1, 1}



{Lタグ、Rタグ}  
左(L)右(R)それぞれの方向に  
データがあれば0、なければ1

図30

【図31】



マーク

図31

【図32】

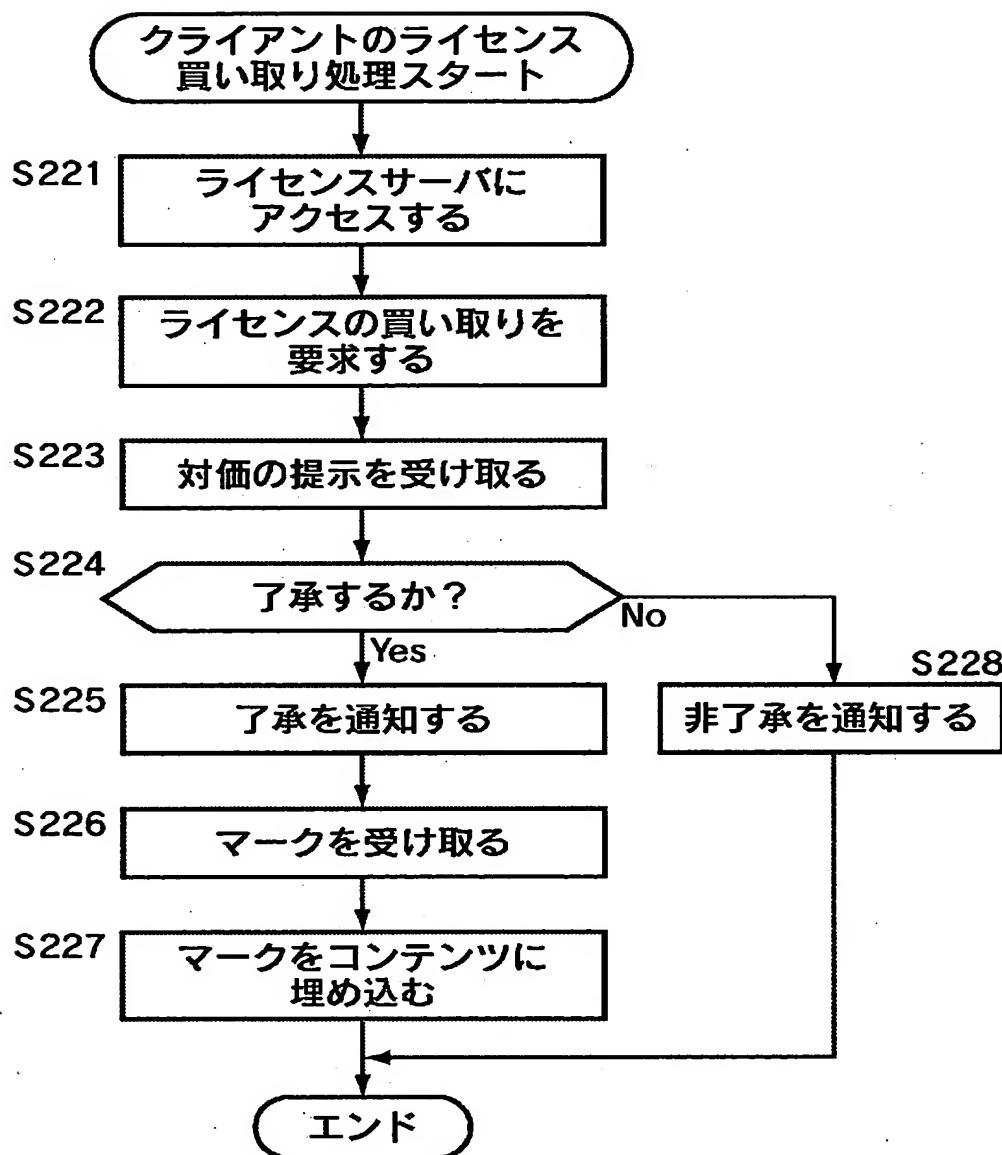


図32

【図33】

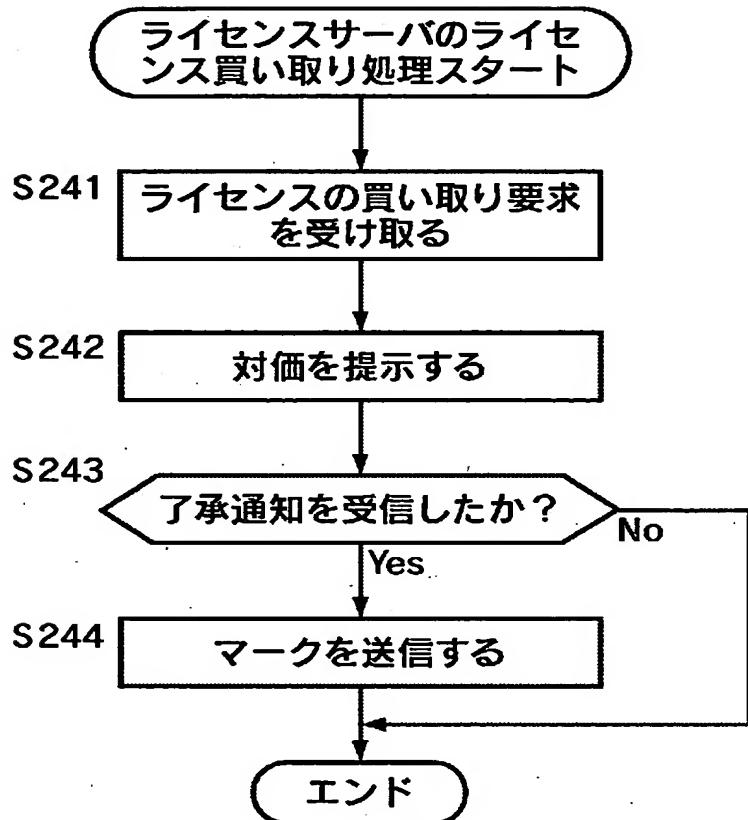


図33

【図34】

$\text{Mark} = \{\text{LeafID}, \text{Own}, \text{Sig}_S(\text{LeafID}, \text{Own})\}$

図34

【図35】

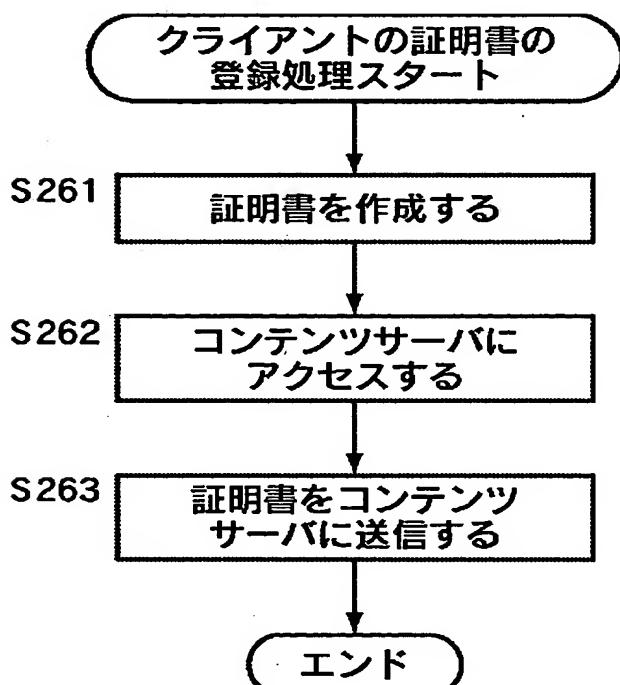


図35

【図36】

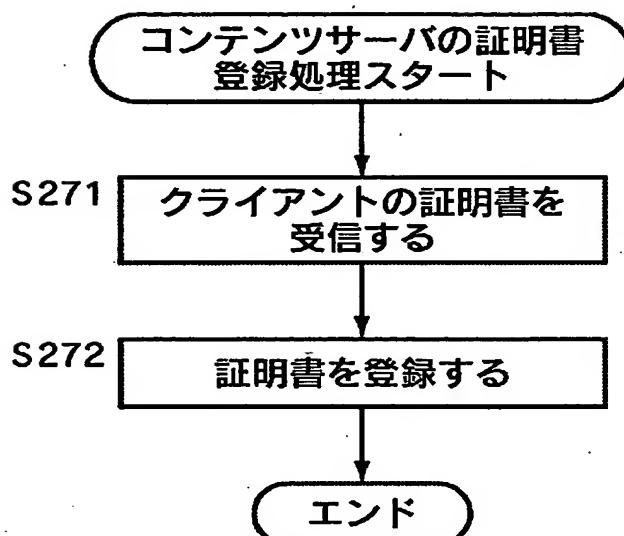


図36

【図37】

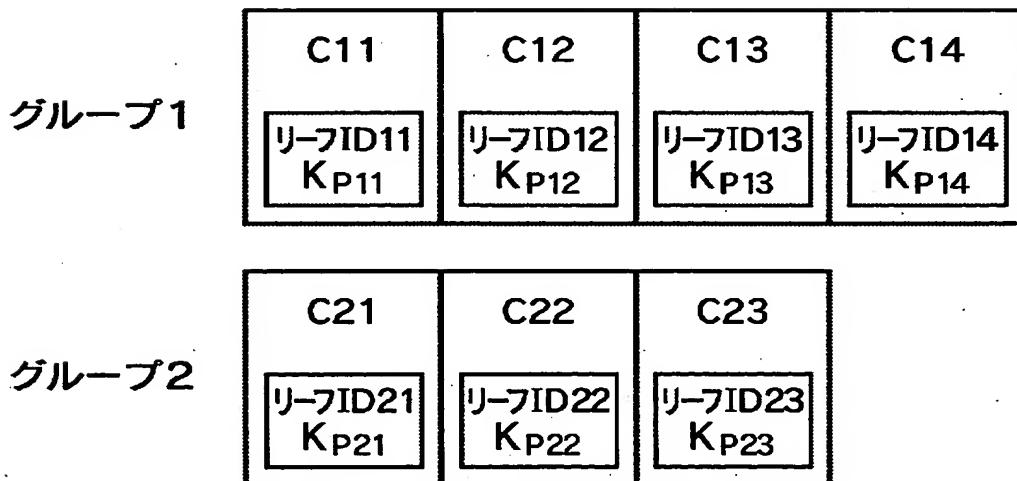


図37

【図38】

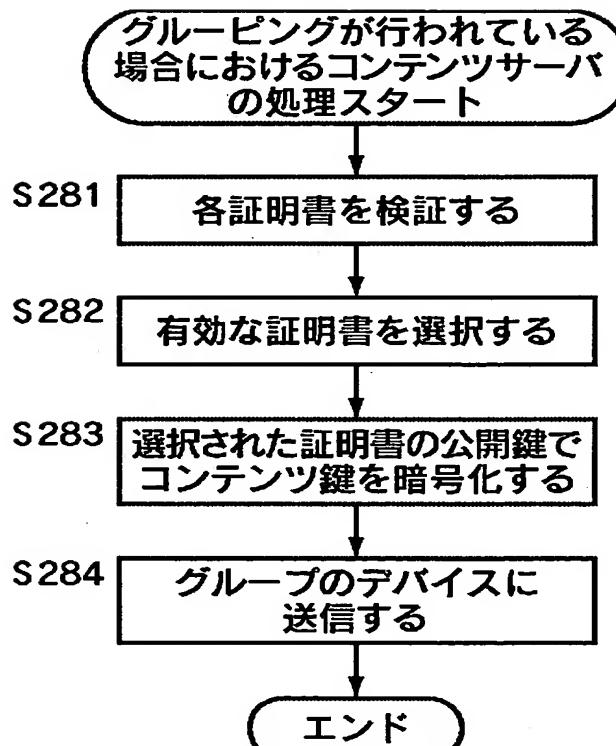


図38

【図39】

$\text{Enc}(K_{P11}, K_C), \text{Enc}(K_{P12}, K_C), \text{Enc}(K_{P13}, K_C)$

図39

【図40】

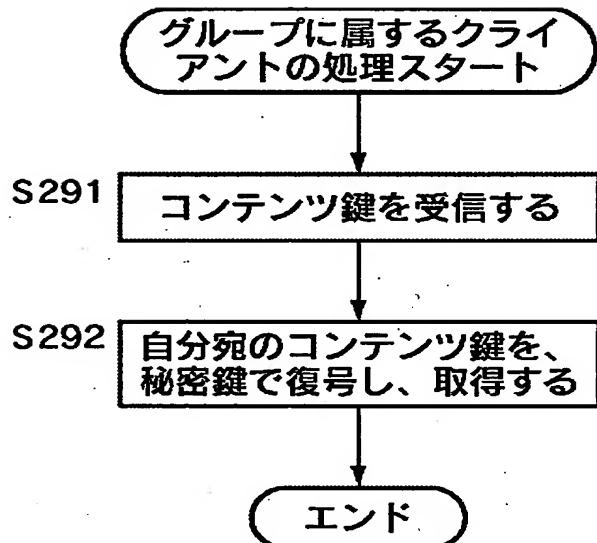


図40

【図41】

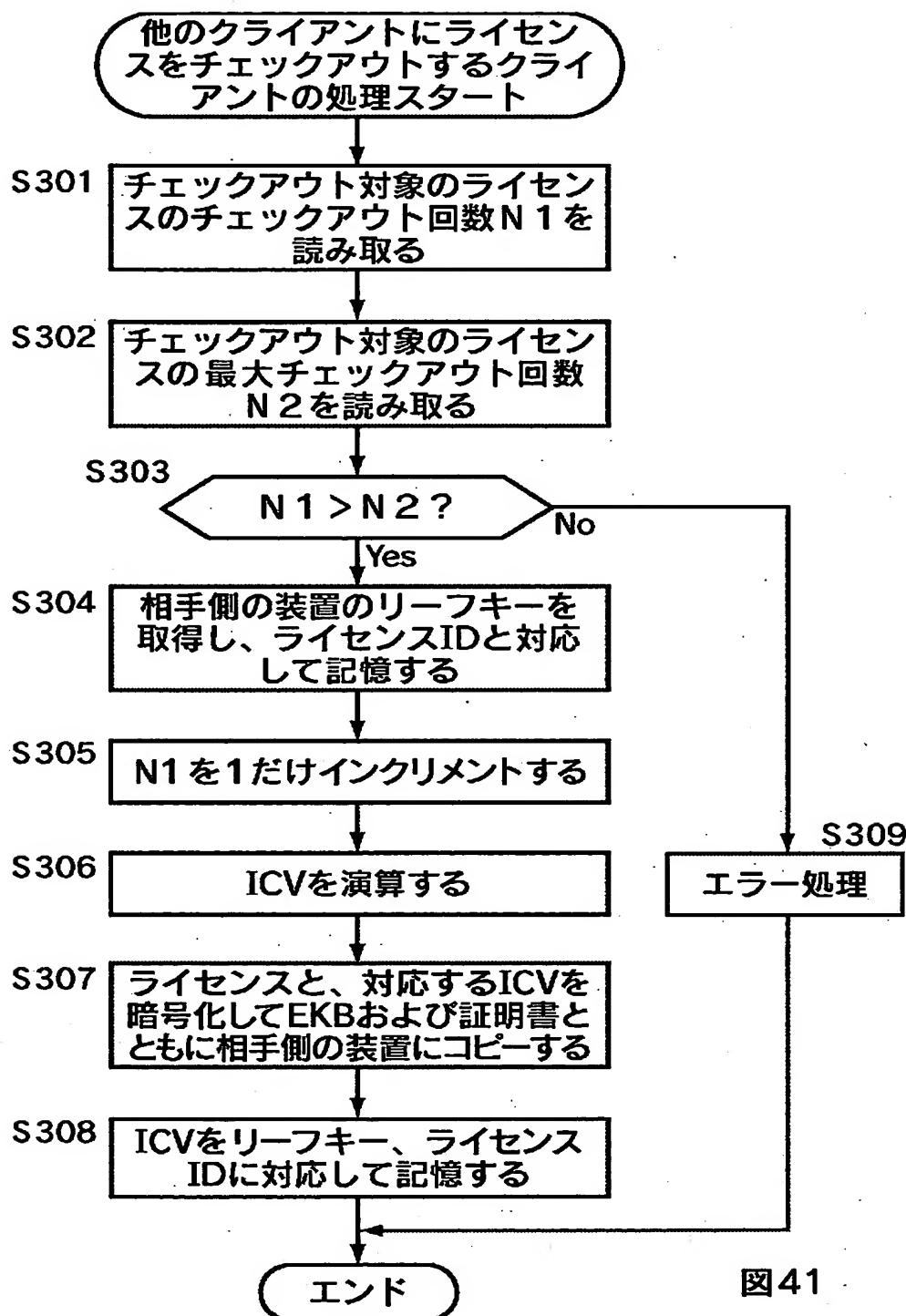


図41

【図42】

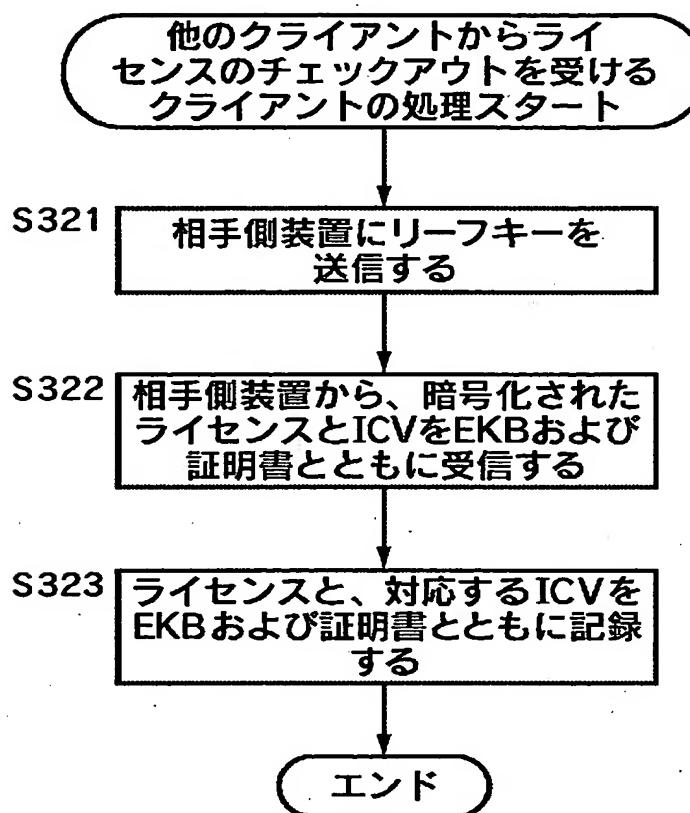


図42

【図43】

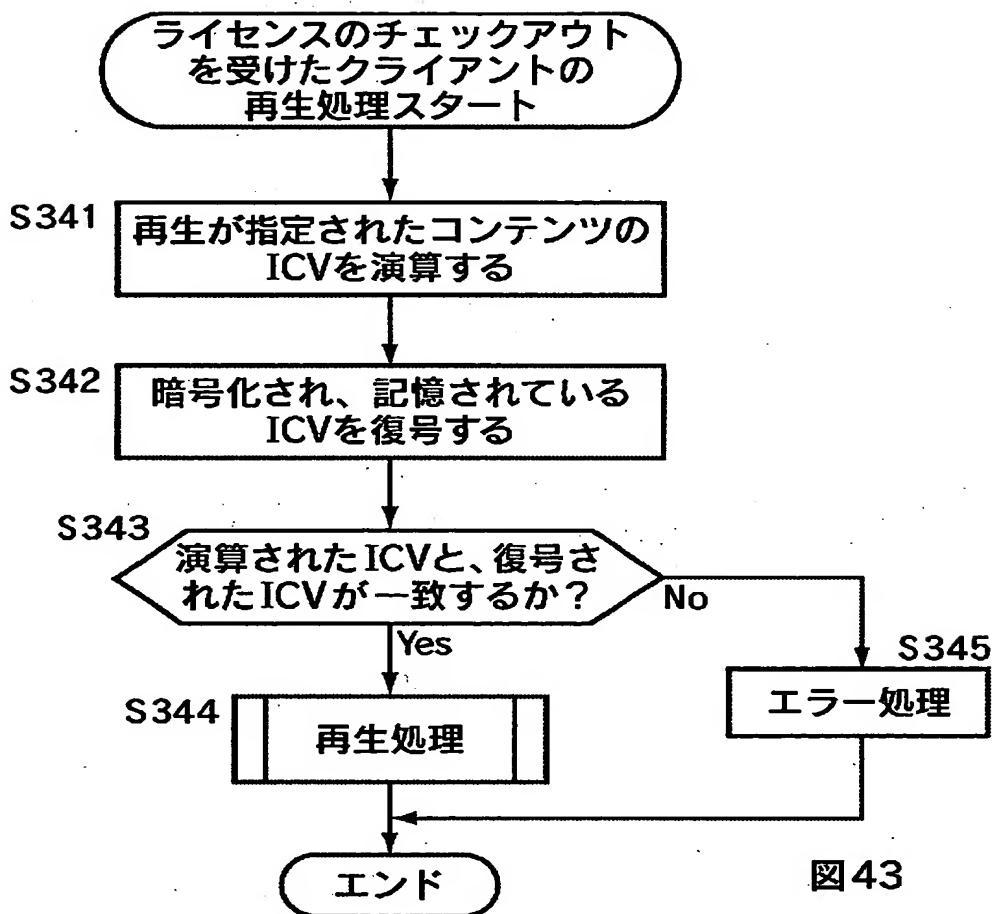


図43

【図44】

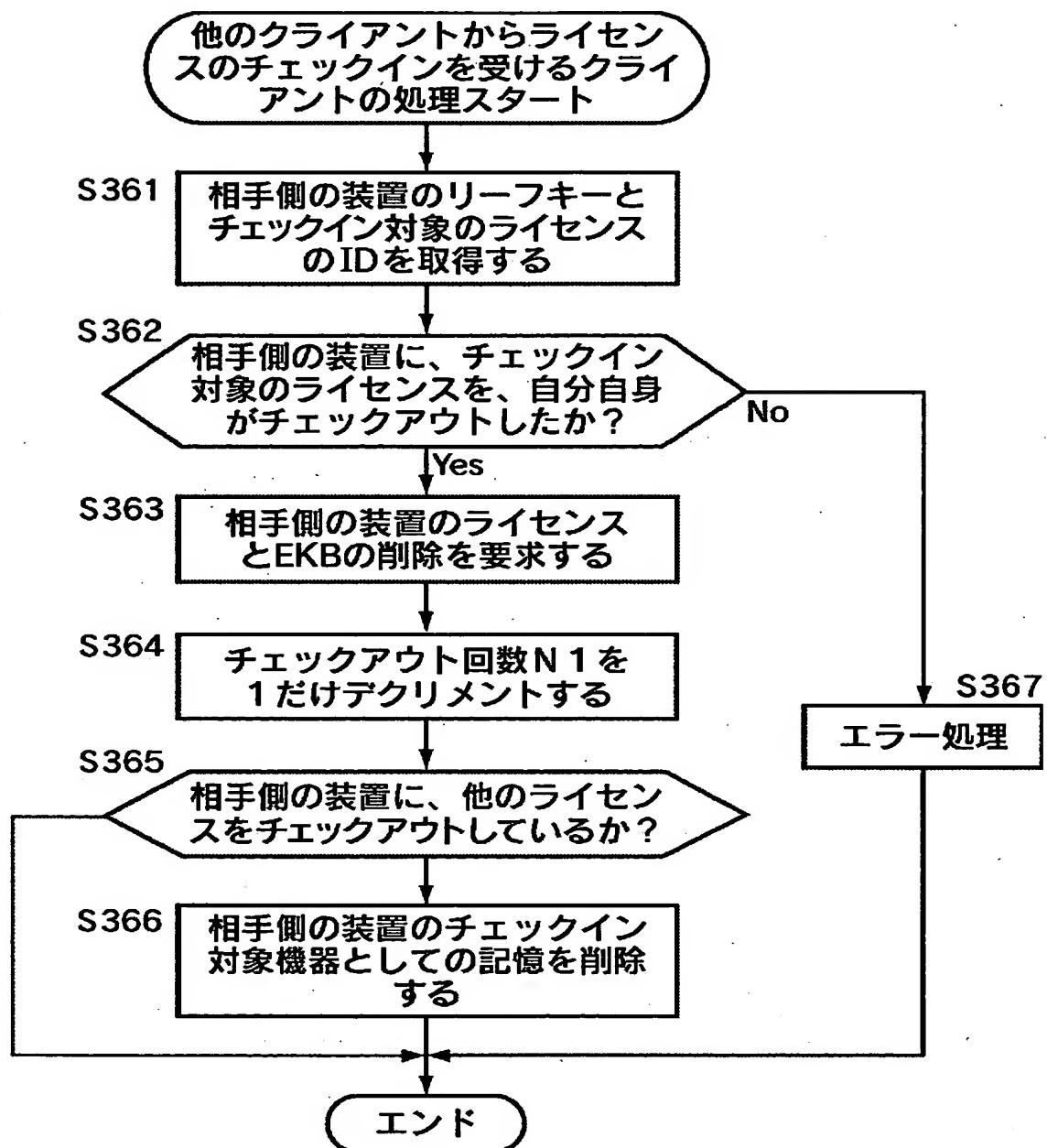


図44

【図45】

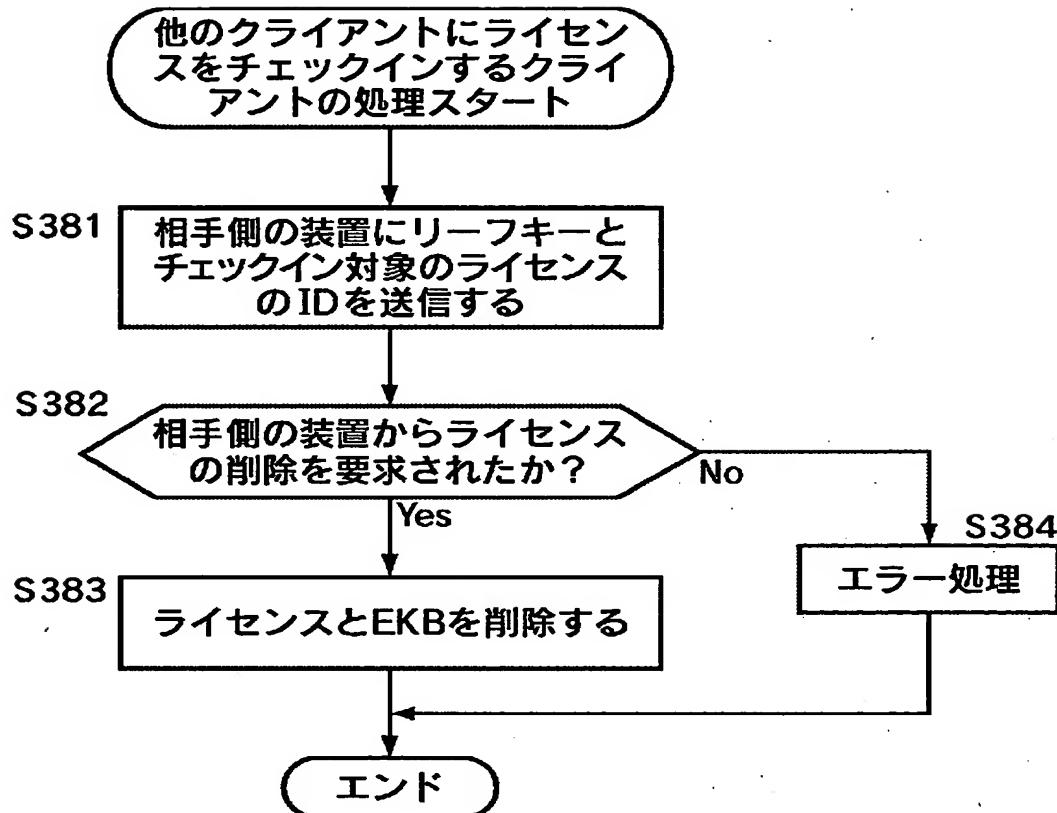
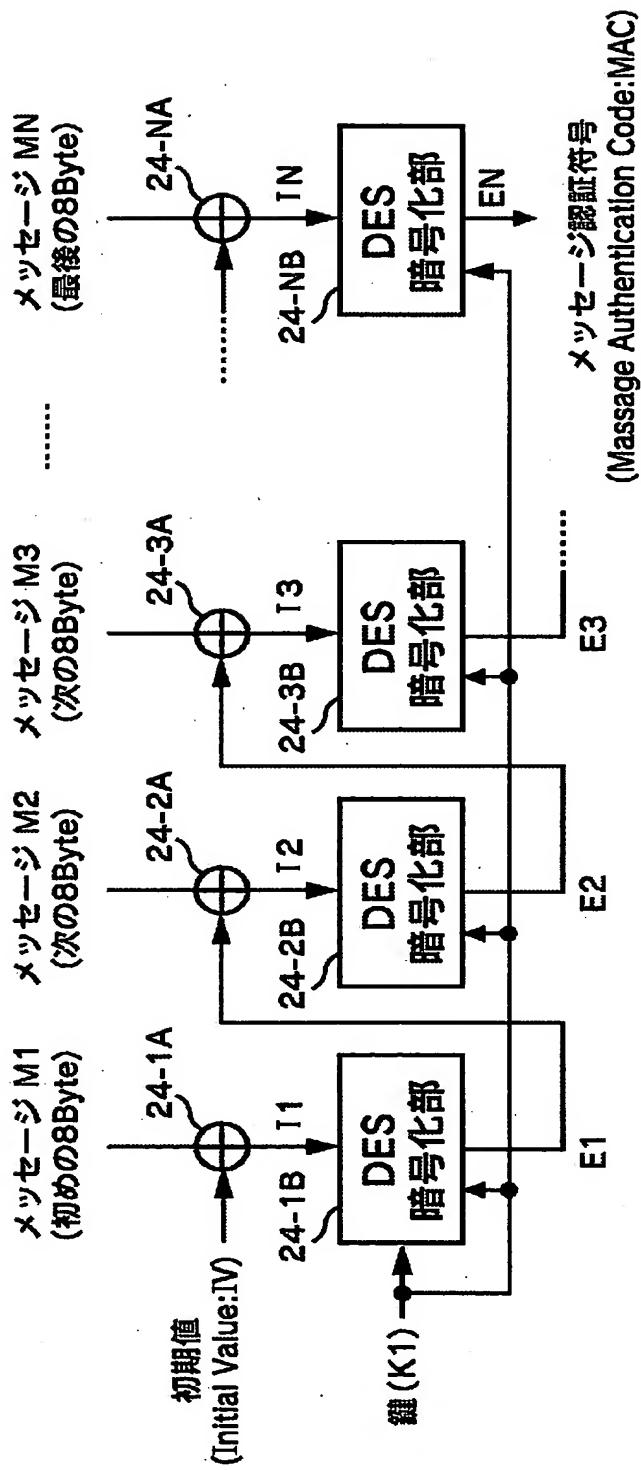


図45

【図46】



⊕ : 排他的論理和処理(8バイト単位)

図46

【図47】

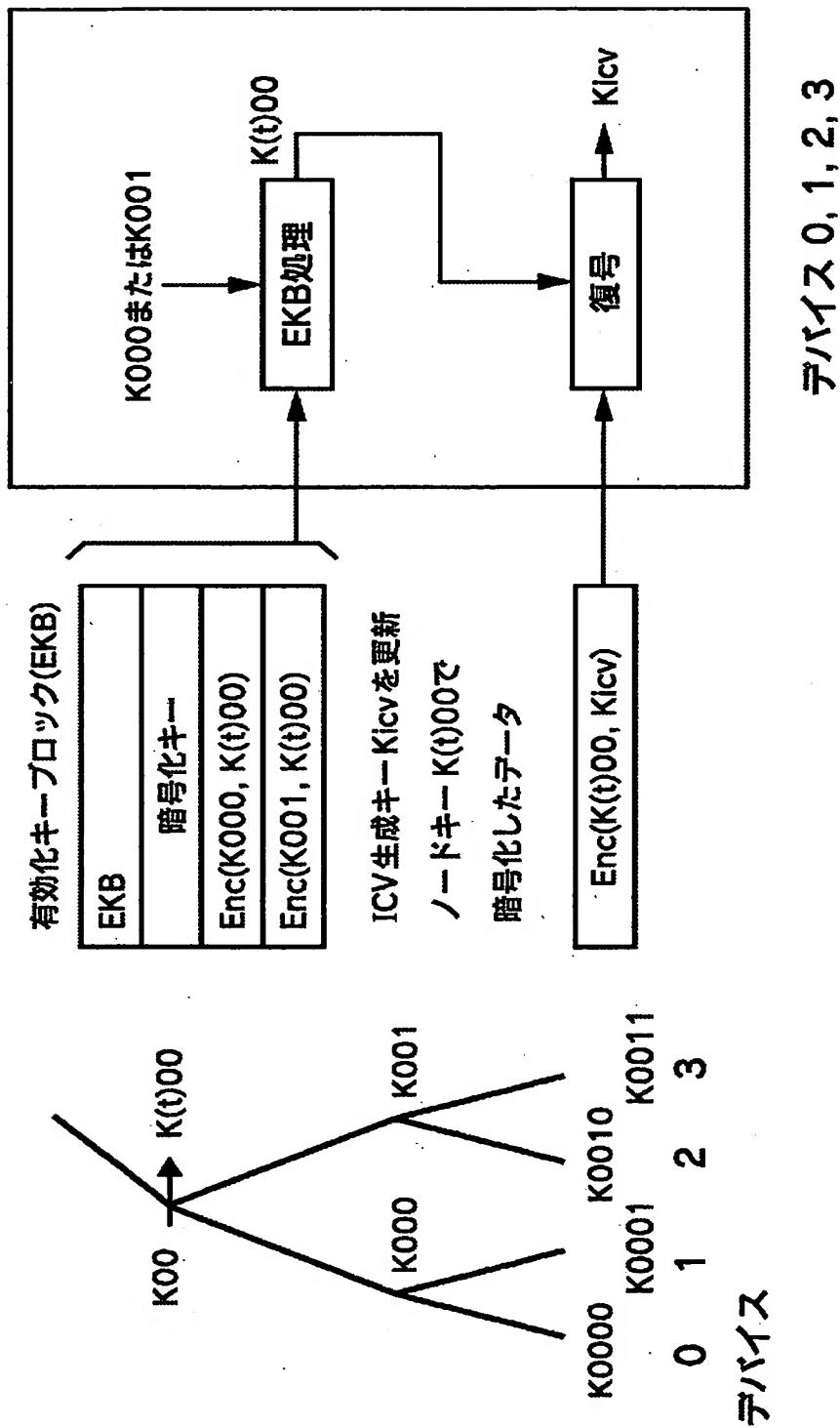


図47

【図48】

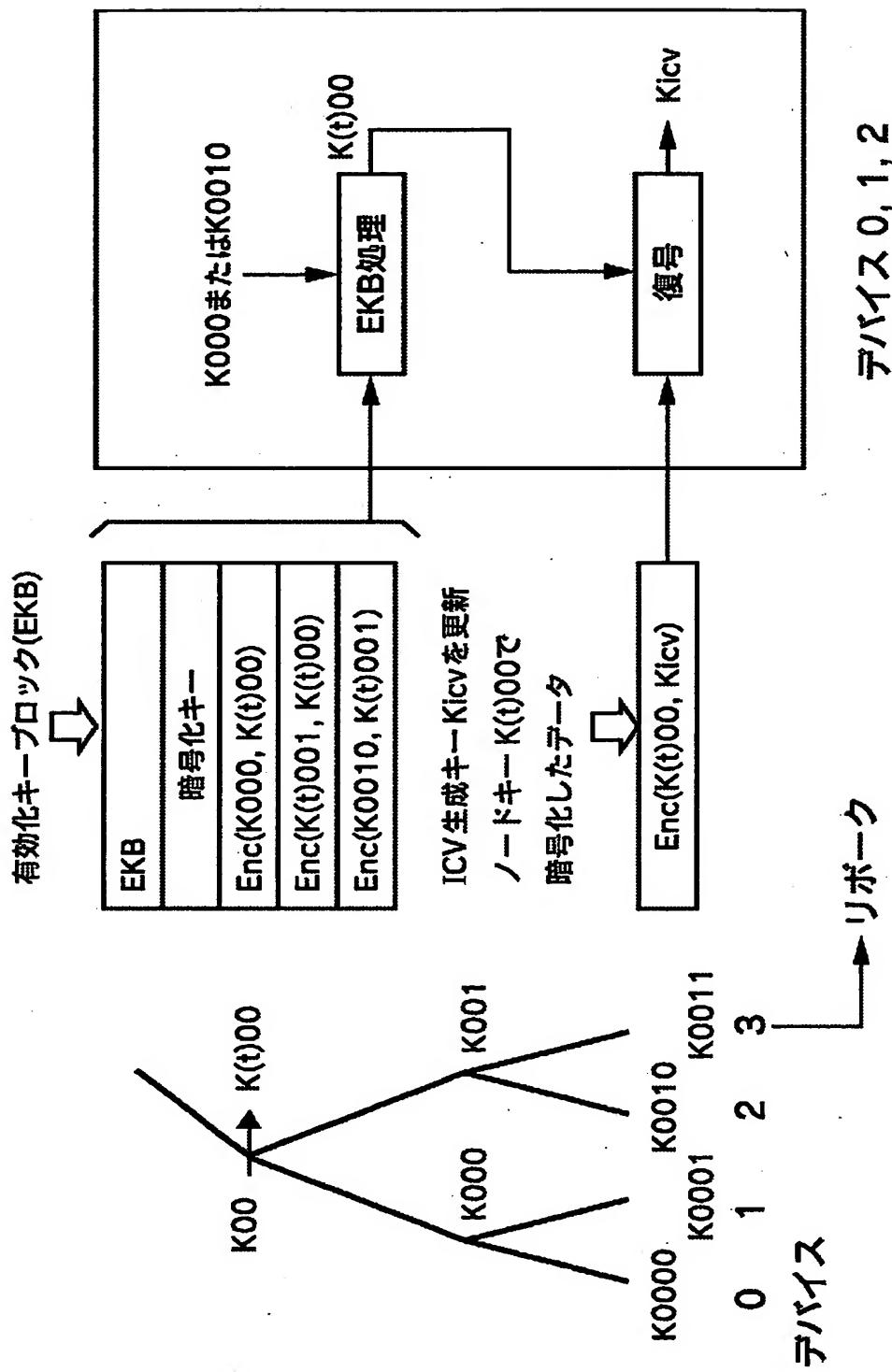


図48

【図49】

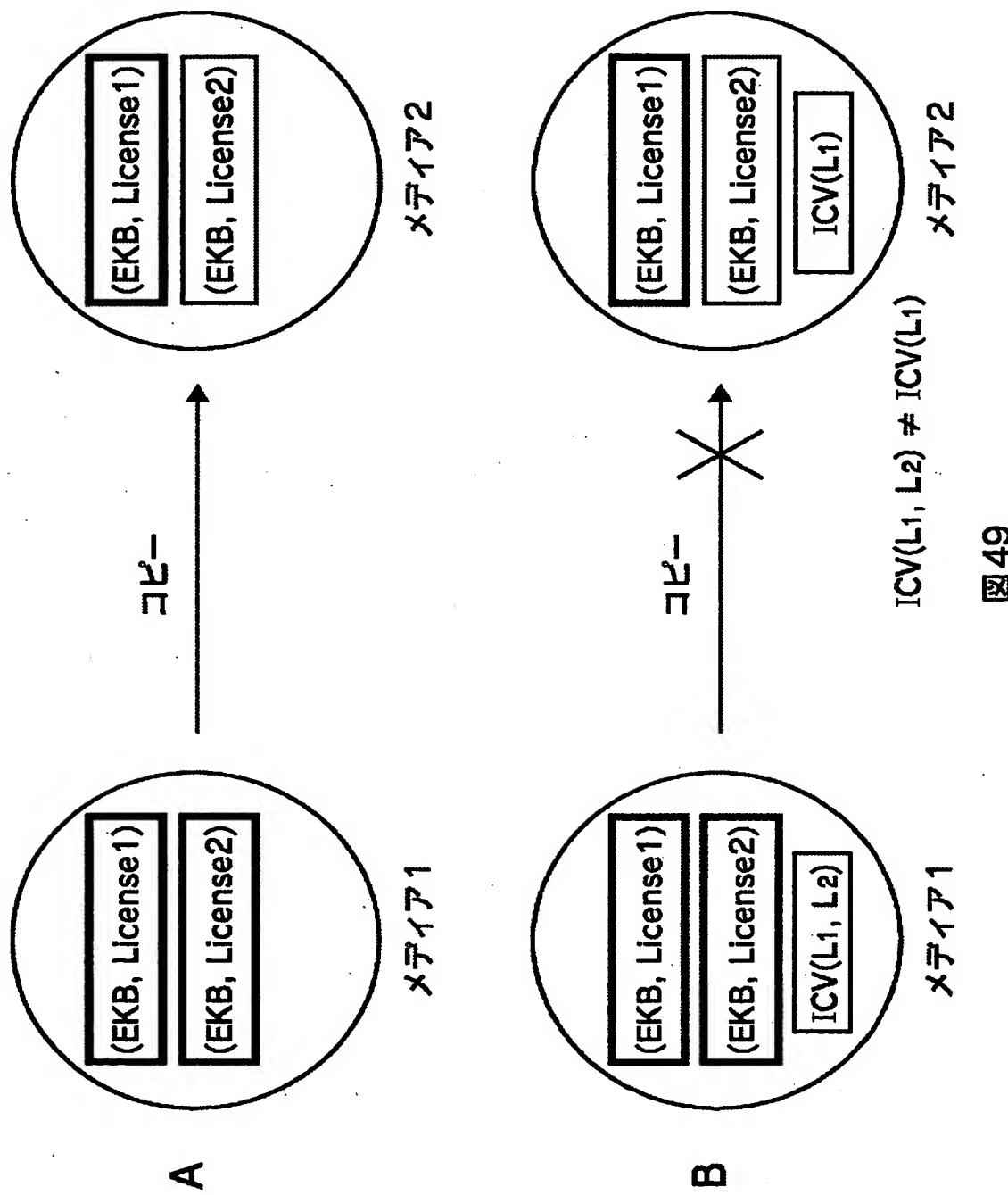
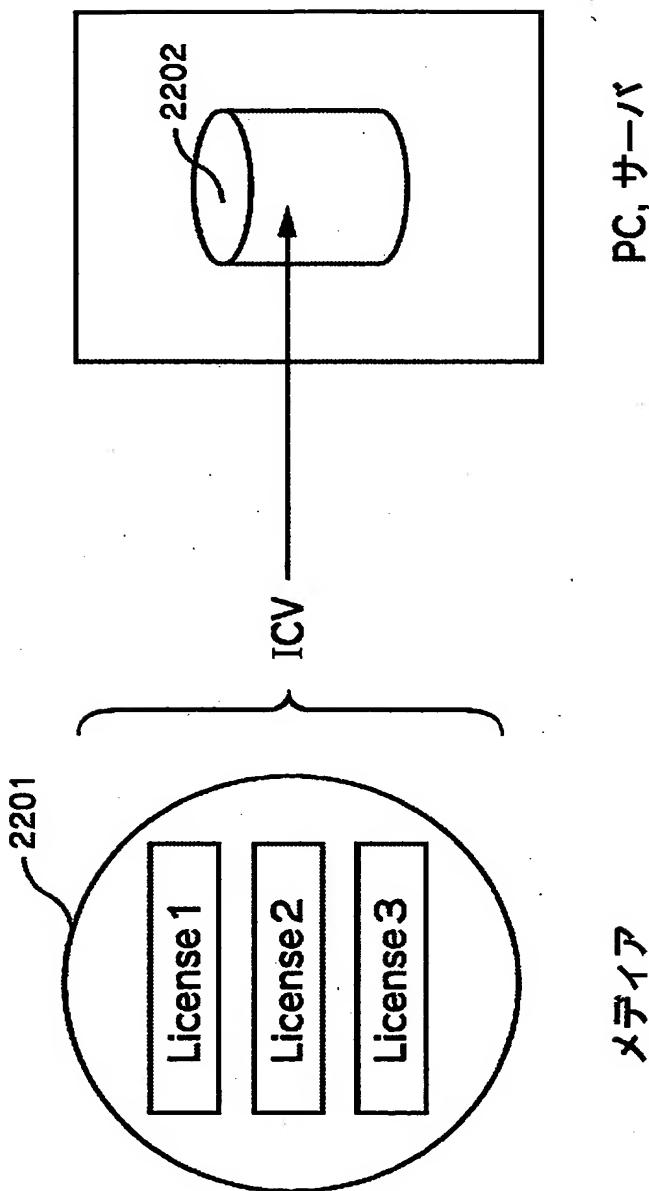


図49

【図50】



PC, サーバ

図50

メディア

【書類名】 要約書

【要約】

【課題】 異なるカテゴリの複数のコンテンツを1つのデバイスで利用できるようにする。

【解決手段】 ツリー構造のノードの中のカテゴリノードに、Tシステムと、所定のデバイス（例えば、メモリスティック）が対応される。Tシステムの下位のノードに対応するコンテンツに対応するデバイスノードキー(DNK1)と、メモリスティックの下位のノードに対応するコンテンツに対応するデバイスノードキー(DNK2)を、同一のデバイスで同時に扱うことができるようとする。

【選択図】 図21

出願人履歴情報

識別番号 [000002185]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都品川区北品川6丁目7番35号

氏 名 ソニー株式会社